EasyAccess2000

Customization Guide

Version 2K.01.36

Cross-Platform Data Compression
Translation • Encryption • Authentication • Digital Signatures

bTrade.com
Deploying e-Business Networks

Contacting Us for Technical Support

All problems relating to EasyAccess2000 should be reported directly to the Help Desk at bTrade.com, 24 hours a day, 7 days a week, by calling **(800) 425-0444** (**972-580-2900** for customers outside North America). Follow the voicemail instructions and press **5** to reach Product Support.

Prime Support Hours: 7 a.m. to 6 p.m. (CST)

After Hours Support: For times outside Prime Support Hours (nights, weekends, and holidays), Technical Support Analysts are on call to respond to Severity 1 issues that can not wait for the next business day.

Please provide questions, suggestions, and feedback on bTrade.com products and documentation by calling **(800) 425-0444** and pressing **5** for Product Support.

bTrade.com
Deploying e-Business Networks

(Formerly COMM-PRESS Technologies, Inc.)
2324 Gateway Drive
Irving, Texas 75063-2743
(972) 580-2900 • (972) 550-7682 (Fax)
www.bTrade.com

Copyrights

This document contains proprietary and confidential material that is the property bTrade.com, Inc. Disclosure outside of bTrade.com, Inc. is prohibited except by license or other confidentiality agreement.

Trademarks

EasyAccess2000, WebAccess2000, SecurePortal2000, SecureAccess2000, SecureManager2000 (formerly KeyManager2000) and Comm-Press2000 are trademarks of bTrade.com.

Windows, Windows 95, Windows 98, Windows NT, MS, Access, SQL Server, and MS-DOS are registered trademarks of Microsoft Corporation.

RSA, RC2, RC4, and BSAFE are registered trademarks of RSA Security, Inc.
RC5 and RC6 are trademarks of RSA Security, Inc.

All other brand and product names are trademarks or registered trademarks of their respective companies.

Table of Contents

Preface

This section provides you with mouse conventions, text notations, and procedures that are common to all bTrade.com technical publications. For software that employs a *graphical user interface* (GUI) and a *command-line interface* (CLI), this guide designates which interface is used to accomplish a procedure.

Mouse Notations

Text	User Mouse Actions
Click	Depress the left mouse button once.
Right-click	Press the right mouse button for these instructions.
Double-click	Depress the left button twice in quick succession.

Typographical Conventions

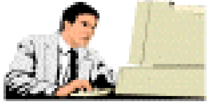
This guide uses typeface changes, symbols, and special icons to set apart information in a structured way that makes it easy for the user to read.

Table 1: What Typeface Changes and Symbols Mean

Typeface or Symbol	Meaning in Paragraph Text, GUI, or Command Line Interface	Examples
italics	Used for: • Document or software titles • *New terms* shown in text • Words that require emphasis	*EasyAccess2000 User Guide* *Digital Encryption Standard* (DES) You *must* be root user to do this.
Bold	Denotes *graphical user interface* (GUI) objects. For example, menu titles, button labels, window names, radio buttons, etc.	When the **Windows NT Security** window displays, click the **Change Password** button.
[**Alt**]+[**F**]	Keyboard keys are enclosed in square brackets and bold font. If the keys must be pressed simultaneously, a plus sign is used in the text.	Press [**Ctrl**]+[**Alt**]+[**Delete**] to log on.
`Bold fixed-width`	Identifies **`user input`** that must be typed exactly	`C:\>`**`add database bhub`**
`Fixed-width`	Identifies command output, including error messages	`>bhub created`
Bold italic fixed-width	Identifies entities you type and are *variables* within command examples that must be supplied by the user. Replace the variable with a real value or name.	`cat` ***file_name***
{ }	Text surrounded by braces (or curly brackets) indicates more than one option. Choices are shown within the braces and separated by a bar {|} divider.	**`boot mode`** {nvram | bootp}
[}	Text surrounded by plain square brackets indicate optional elements in the command examples.	`rm` ***input_file [input_file …]***
…	Horizontal ellipsis indicates you can supply more than one value or parameter for preceding item(s)	`rm` ***input_file [input_file …]***

Reader Alerts

This document presents ideas (timesavers), notes, cautions, warnings, and advice to highlight information of direct importance to you:

Icon	Reader Alert Description
Idea	**Idea** – Provides a specific procedure or describes where to obtain more information that helps get your job done faster.
Take Note	**Note** - Highlights special information that is pertinent to the primary discussion. This information is important enough to you, that it is set off from normal text, and called to your attention.
CAUTION	**Caution** - Identifies information that is critical to the operation or procedure and is necessary to *prevent* loss of data.
	**Contact bTrade.com Product Support for additional information.**

Viewing Online bTrade.com Documentation

The online bTrade.com documentation can be read using the Adobe Acrobat™ Reader. (If you do not have this software, you can download it from Adobe's website - http://www.adobe.com/.) The Adobe Portable Document File (PDF) displays the bTrade.com user guides in full color and acts similar to an online help system.

With the online documentation guide you can:

- Control the size of the displayed information
- Print all or a portion of the user guide.
- Find a specific topic using full-text search procedure
- Use bookmarks and hyperlinks to swiftly navigate among the pages.

Procedure A: Displaying the Online User Guide

1. Double-click the user guide file (file extension `.pdf`) with Microsoft Windows Explorer® or use the Acrobat Reader program's **Open** command found under the **File** menu.
2. Press [`Ctrl`]+[`M`] keys to access the **Zoom To** dialog box.
3. Type a value for the **Magnification** you desire and click **OK**. The user guide page displays at the specified magnification.

Procedure B: Viewing an Online User Guide with Bookmarks

1. Choose the **Show Bookmarks** command from the **Windows** menu. The bookmarks display as an “interactive” table of contents.
2. Click the **Bookmark** for the user guide section you want to view. The Bookmark’s page and location display in the **Acrobat** window.

Procedure C: Printing the Online User Guide

1. Choose the **Print** command from the **File** menu or press [`Ctrl`]+[`P`] keys to access the **Print** dialog box.
2. Select the printer and specify the number of copies to print.
3. Type the page numbers (starting and ending) in the **From** and **To** text fields.
4. Click the **Print** button.

To print the online user guide, you must have Adobe Acrobat Reader or the full Acrobat product installed.

Take Note

This online bTrade.com guide has page numbers that begin with **one** as the title page. This helps the reader to print the pages accurately using the above Adobe Acrobat Reader procedure.

Procedure D: Searching the Online User Guide

1. Choose the **Find** command from the **File** menu or press [`Ctrl`]+[`F`] keys to access the **Find** dialog box.
2. Type the word (or words) to search for in the text field.
3. Click the **Find** button.
4. Press [**Ctrl**]+[**G**] to find the next occurrence of the search words.

Technical Support

All problems relating to EasyAccess2000 should be reported directly to the Help Desk at bTrade.com, 24 hours a day, 7 days a week, by calling **(800) 425-0444** (**972-580-2900** for customers outside North America). Follow the voicemail instructions and press **5** to reach Product Support.

Prime Support Hours: 7 a.m. to 6 p.m. (CST)

After Hours Support: For times outside Prime Support Hours (nights, weekends, and holidays), Technical Support Analysts are on call to respond to Severity 1 issues that can not wait for the next business day.

Please provide questions, suggestions, and feedback on bTrade.com products and documentation by calling **(800) 425-0444** and pressing **5** for Product Support.

FAX	1- (972) 550-7682
E-mail	help@btrade.com
Address	2324 Gateway Drive Irving, Texas 75063-2743
Website	www.bTrade.com

Creating Diagnostic Information

Technical Support may request that you collect additional information for problem determination. Specifically, these changes order EasyAccess2000 to create detailed log information:

1. Shutdown the EasyAccess2000 GUI application.
2. Edit the `easyacc.ini` file in the `easyacc6` subdirectory folder.
3. Find the `Identify` section within the `easyacc.ini` file.
4. Change these keywords so that the keyword values equal `-6`:

```
LOG_MEM=-6
LOG_INI=-6
LOG_XFER=-6
LOG_FTP=-6
LOG_EASYACC=-6
LOG_THREAD=-6
```

5. These files store the logging information and bTrade.com Product Support may request that you email the files to our support center. (See Figure 1 regarding EasyAccess2000 directory structure.)

```
EASYACC6\easyacc.ini
EASYACC6\exfer.ini
EASYACC6\baseout.msg
EASYACC6\baseout.ms1
EASYACC6\baseout.ms2
EASYACC6\temp\compress.log
EASYACC6\temp\decomp.log
EASYACC6\temp\eaftp.log
EASYACC6\temp\list.fil
EASYACC6\temp\temp.fil
EASYACC6\temp\audit.log
EASYACC6\temp\eaxfer.log
```

6. To restore the logging data collection keywords to their normal (production) values, change the keywords to:

```
LOG_MEM=N
LOG_INI=N
LOG_XFER=N
LOG_FTP=6
LOG_EASYACC=N
LOG_THREAD=N
```

Introduction

The *EasyAccess2000 Customization Guide* describes the additional capabilities of the EasyAccess2000 application. It demonstrates how to use EasyAccess2000 to perform secure batch file transfers and how to include EasyAccess2000 in user written programs. Refer to the *EasyAccess2000 User Guide* for detailed instructions about using the EasyAccess2000 graphical user interface (GUI).

Assumptions

This guide assumes that the reader has a general understanding of:

- The workstation's operating system
- Command-line interface usage
- Running batch programs and scripting

Customization Guide Sections

Preface is a set of standard instructions to help the reader use this documentation.

Section 1, Introduction, introduces the reader to specific sections within the guide and tells you where to find additional assistance. It describes the basic EasyAccess2000 functionality and the files that control EasyAccess2000 key features.

Section 2, Configuring EasyAccess2000, provides the major steps necessary to configure EasyAccess2000 for key computing operating systems.

Section 3, EasyAccess2000 Keyword Reference, lists keywords and arguments, describes the appropriate syntax, and provides command file examples for the command-line interface.

Section 4, Using EasyAccess2000 Applications, provides examples of using the EasyAccess2000 command-line interface and utility applications.

Section 5, Using the IEBASE.EXE Application, explains how EasyAccess2000 can be used to interpret BASEIN.MSG command files and perform batch transmissions with *FedEx Net* (FEDEXNET) and *IBM Global Network Information Exchange* (IGN-I/E).

Section 6, Scheduling Automated Data Transfers, explains how to run EasyAccess2000 to access multiple mailboxes/user IDs without operator intervention.

Section 7, Easyacc.ini File Reference, an A-Z reference that describes the keywords and the easyacc.ini file structure used to provide flexible configurations for EasyAccess2000 networks.

Section 8, Glossary, defines some of the more cryptic (pun intended) terminology found in this EasyAccess2000 Customization Guide.

Section 9, Index, cross-reference of keywords and concepts presented in this guide.

Related Documents and Standards

This section describes documentation that contains information about subjects related to procedures discussed in the *EasyAccess2000 Customization Guide*.

Table 2: Related Documents and Internet Standards

Document Number	Title
	EasyAccess2000 User Guide
	EasyAccess2000 Online Help – context-sensitive help
	Comm-Press2000 User Guide – describes some of the advanced encryption, compression, and decryption options used.
RFC-959	File Transfer Protocol. J. Postel, J.K. Reynolds. Oct-01-1985. (Obsoletes RFC-765) (Updated by RFC-2228, RFC-2640) (Status: STANDARD)
RFC-1113	Privacy enhancement for Internet electronic mail: Part I - message encipherment and authentication procedures. J. Linn. Aug-01-1989. (Obsoleted by RFC-1421).
RFC-1421	Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures. J. Linn. February 1993. (Obsoletes RFC-1113) (Status: PROPOSED STANDARD)
RFC-1767	MIME Encapsulation of EDI Objects. D. Crocker. March 1995. (Status: PROPOSED STANDARD) **Note:** For `EDI_INT` network-style, use MIME Content-Types defined in RFC-1767 for EDI data (EDI-X12, EDIFACT, or EDI-Consent types).
RFC-2228	FTP Security Extensions. M. Horowitz, S. Lunt. October 1997. (Updates RFC0959) (Status: PROPOSED STANDARD)
RFC-2640	Internationalization of the File Transfer Protocol. B. Curtin. July 1999. (Updates RFC-959) (Status: PROPOSED STANDARD)

Where to Find More Assistance

Table 3: Where to get More Assistance

Where to find Assistance	To get assistance with this situation
bTrade.com Product Support	Need *dynamic link libraries* (DLLs) that support Encryption features of Comm-Press2000 (version 4.4) to de-compress the data transfers.
EasyAccess2000 Online Help	Set-by-step procedures using the EasyAccess2000 graphical user interface.
EasyAccess2000 User Guide	Set-by-step procedures using the EasyAccess2000 graphical user interface.

Functionality Overview

When used in a server configuration, EasyAccess2000 creates secure, batch data transfers using:

- *File Transfer Protocol* (FTP)
- *Simple Mail Transfer Protocol* with *Post Office Protocol 3* (SMTP/POP3)

With bTrade.com's SecurePortal2000 application, it can use additional communications protocols like:

- *Electronic Data Interchange-Internet Integration-Applicability Statement 1* (EDI-INT AS1) that uses *Multipurpose Internet Mail Extension* (MIME) and SMTP Internet standards
- *Applicability Statement 2* (AS2) using MIME and *Hypertext Transfer Protocol* (HTTP) standards
- *Gas Industry Standards Board* (GISB) that uses HTTP protocol with *Pretty Good Privacy* (PGP)

EasyAccess2000 uses compression and security of local, non-transport requirements. Several configuration, command, and message files control the EasyAccess2000 application operation. These files permit companies to schedule automated data transfers to fit their business requirements.

EasyAccess2000 requires a *Transmission Control Protocol/Internet Protocol* (TCP/IP) connection to a network that can access your *trading partner*'s computer server (for example, via the Internet) to perform data transfers. EasyAccess2000 is dependent on the hub-trading partner or your *Value Added Network* (VAN) for its initial data transmission configuration. The hub-trading partner or VAN also manages the encryption keys with other security information required for secure data transmission.

Trading Partner Configuration Files

Table 4: EasyAccess2000 Trading Partner/VAN Configuration Files

File Name	Description of File Contents
`easyacc.ini`	**Profile and configuration information:** • Hostname or TCP/IP address of host server • User ID for user logon to a host server • Password (encrypted) for user logon to a host server • File specifications for files to be sent or received • Comm-Press2000 file transfer compression and decompression options • List of *Transfer* names for *Stored Transfers* or *Batch mode* • Adjust the amount of log information (see *EasyAccess2000 User Guide*)
`exfer.ini`	Predefined stored transfer operation instructions
`bexfer.ini`	Predefined operations for the `IEBASE` utility program (used by the IGN-I/E network).
`Tpaddrss.ini`	Information about trading partners

Caution: Do not change the contents of these EasyAccess2000 configuration files unless specifically instructed by bTrade.com Product Support personnel.

Security Runtime Files

Encryption keys and security configuration data is created by the SecureManager2000 application and stored in a group of files collectively known as the *Security Runtime Files*. These files are created during the customization process after the EasyAccess2000 software installation.

Table 5: EasyAccess2000 Security Runtime Files

File Name	Description of File Contents
`alias.tbl`	**Alias lookup table** – records that define alias (other names) for trading partner networks.
`cert.fil`	**Certificate** - public keys of all trading partners who exchange secure data.
`cplookup.tbl`	**Comm-Press2000 lookup table** - records that define the Comm-Press2000 security options being used between trading partners.
`private.fil`	**Private key** - private keys of local security participants that wish to send secure data to outside trading partners
`private.key`	**Permanent key file** - created by `GENKEYS` utility, it must be retained as part of the request for a digital certificate from a trading partner. When the trading partner or SecureManager2000 issues security runtime files, information from this file is used during the **Import** process.
`symkey.fil`	**Symmetric key** - secret keys of local security participants that wish to send secure data to outside trading partners using secret key cryptography.

EasyAccess2000 exchanges data with the server by running data transfers. Pre-defined or stored transfers are contained in the `exfer.ini` file. You can run stored transfers by using their names as parameters when executing the EasyAccess2000 program. One-time or ad-hoc data transfers can also be created and run via EasyAccess2000 *command-line interface* (CLI).

Command-Line Interface Procedures

Unless otherwise stated, each procedure in this document is using the command-line interface (CLI). EasyAccess2000 graphical user interface (GUI) procedures are documented in the *EasyAccess2000 User Guide*.

Configuring EasyAccess2000

Operating Systems for EasyAccess2000

EasyAccess2000 provides a *command-line interface* (CLI) and a *graphical user interface* (GUI) for several different computer operating systems. For the different Microsoft Windows operating systems, several *dynamic link libraries* (DLLs) are available. For each major operating system type (Windows, Unix, DEC, AS/400, and MVS there is a different installation and configuration procedure..

Table 6: EasyAccess2000 Operating Systems Applications

Operating System	Application Name for Interface		DLLs For Windows Operating Systems
	CLI	GUI	
Windows - 95/98/2000 Client	`ea2kw95c`	`ea2kw95`	`ea2kw95.dll`
Windows – NT/2000 Server	`ea2kwntc`	`ea2kwnt`	`ea2kwnt.dll`
UNIX - AIX 4.1 or higher	`ea2kaixc`	`ea2kaix`	
UNIX - HP-UX 10.01 or higher	`ea2khpuxc`	`ea2Khpux`	
UNIX - Sun Solaris 2.6 or higher	`ea2ksunc`	`ea2ksun`	
UNIX - SCO 3.2	`ea2kscoc`	`ea2ksco`	
DEC - Alpha VMS 7.2	`ea2kvmsc`		
DEC - Tru64 4.0	`ea2ktru64c`		
AS/400 – OS/400 V3R7M0 or higher	`ea2k400c`		
IBM - MVS 4.3, OS/390 1.2 or higher	`ea2kmvsc`		

A blank denotes no application or library available for this operating system.

For Windows 95/98/NT/2000 and UNIX Systems

Overview

To configure EasyAccess2000 for the Windows or UNIX operating systems you need to:

1. Generate the EasyAccess2000 Encryption Keys
2. Send the certificate request file to the hub-trading partner.
3. Receive the Security Runtime Files from the hub-trading partner.
4. Install the Security Runtime Files for a non-SSL Network
   or
   Install the Security Runtime Files for IGN-I/E SSL Network

There are several ways you can configure the EasyAccess2000; based upon your preferences, software available, and the network type. Many customers use the EasyAccess2000 *graphical user interface* (GUI) to perform the configuration operations. In some situations you may be unable to (do not have Motif software) or prefer to use the *command-line interface* (CLI)

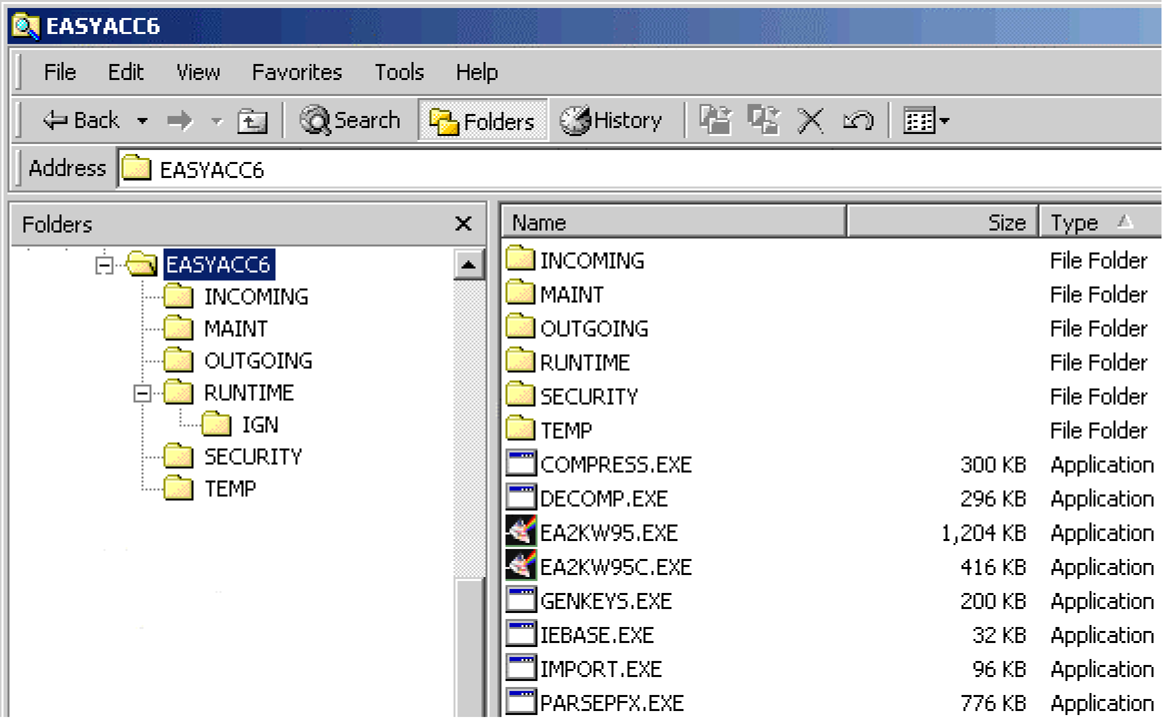
Table 7: Configuration Software to Use

Configuration Action	Operating System	Can Use GUI	Must or Prefer to Use the CLI
Generate Encryption Keys	Windows	Yes	Use `ea2kw95c` or `ea2kwntc`
	UNIX	Yes	Use `genkeys` utility
Install Security Runtime Files	Windows	Yes	Use `ea2kw95c` or `ea2kwntc`
(non-SSL network)	UNIX	Yes	Use `Import` utility
Install Security Runtime Files	Windows	n/a	Use `parsepfx` utility
(IGN-I/E SSL network)	UNIX	n/a	Use `parsepfx` utility

File Directory Structure

The EasyAccess2000 application relies on its file subdirectory structure to accomplish its tasks. The directory structure for EasyAccess2000 on a workstation displays similar to the next figure.

Figure 1: EasyAccess2000 File Directory Structure



Name	Size	Type
INCOMING		File Folder
MAINT		File Folder
OUTGOING		File Folder
RUNTIME		File Folder
SECURITY		File Folder
TEMP		File Folder
COMPRESS.EXE	300 KB	Application
DECOMP.EXE	296 KB	Application
EA2KW95.EXE	1,204 KB	Application
EA2KW95C.EXE	416 KB	Application
GENKEYS.EXE	200 KB	Application
IEBASE.EXE	32 KB	Application
IMPORT.EXE	96 KB	Application
PARSEPFX.EXE	776 KB	Application

CAUTION

Do not delete or move any of the EasyAccess2000 file subdirectories. Refer to the *EasyAccess2000 User Guide* for specific installation instructions. EasyAccess2000 relies on this file subdirectory structure to accomplish its tasks.

Table 8: EasyAccess2000 Applications and Utilities

Application / Utility Name	Application or Utility Functionality
`EA2KW95.EXE`	EasyAccess2000 Windows graphical user interface (GUI)
`EA2KW95C`	EasyAccess2000 Windows command-line interface (CLI)
`GENKEYS.EXE`	A command-line interface utility to generate public/private key pair and store it in `SECURITY` folder.
`IMPORT.EXE`	A command-line interface utility to install the Security Runtime Files received from the hub-trading partner.
`PARSEPFX.EXE`	A command-line interface utility to install the digital certificates received for a *IBM Global Network Information Exchange* (IGN-IE) *Secure Socket Layer* (SLL) network.
`COMPRESS.EXE`	Compresses data for transfer using the Comm-Press2000 keywords.
`DECOMP.EXE`	Decompresses received data using the Comm-Press2000 keywords.
`IEBASE.EXE`	A utility to interpret **`BASEIN.MSG`** command files and perform batch transmissions with *FedEx Net* (FEDEXNET) and *IBM Global Network Information Exchange* (IGN-IE). A user can send and receive to multiple mailboxes, plus perform multiple logons during a single session.

Procedure A: Generating EasyAccess2000 Encryption Keys

Before EasyAccess2000 can transmit secure data, you must generate encryption keys and have them certified by the hub-trading partner or get them from your trading partner/VAN. Microsoft windows users can generate the keys using the EasyAccess2000 GUI, which is discussed in detail in the *EasyAccess2000 User Guide*. UNIX users that do not have Motif, and Windows or Unix users that prefer to use the command-line interface, can use the `GENKEYS` utility program to generate the encryption keys.

Generating encryption keys

1. Move the current directory designation the top-level EasyAccess2000 directory folder. Use the **`cd`** command (UNIX or Windows batch command).
2. Type **`genkeys`** at the command prompt and press [**Enter**] to start the utility.
3. When the `GENKEYS` utility prompts for random input data, type several lines of random characters to create encryption keys that are difficult to unscramble.
4. Press [**Enter**] on a blank line to complete the random entry. The `GENKEYS` utility writes your `private.key` and `cert.req` files to the EasyAccess2000 `Security` subdirectory.

The `private.key` file created by `GENKEYS` utility is a permanent key file and must be retained on the user's workstation. The `cert.req` file contains the portion of the key that must be certified by the hub-trading partner and is transmitted to the hub.

Procedure B: Sending a Certificate Request

Most users can transmit the `cert.req` file to the hub by calling EasyAccess2000 and running the `SEND CERT REQ` stored transfer. Specific instructions for sending the `cert.req` file are provided by the hub-trading partner. Once the hub account receives the request, it is imported and certified. After certification, the run-time files are exported and sent to the user's mailbox.

Procedure C: Receiving Security Runtime Files from Trading Partner

The runtime files are distributed from the hub compressed and encrypted. Most users can receive the compressed file by executing EasyAccess2000 and running the `RECEIVE RUNTIMES` stored transfer. Specific instructions for receiving the compressed and encrypted file are provided by the hub-trading partner.

Procedure D: Installing Security Runtime Files (non-SSL Networks)

The hub issues the Security Runtime Files required to exchange secure data between this workstation and the trading partner. To complete the security configuration, you must install the Security Runtime Files generated by the hub-trading partner. This is accomplished by using the EasyAccess2000 graphical user interface application. Windows users can use the GUI that is discussed in the *EasyAccess2000 User Guide*. UNIX users that do not have Motif software, and Windows or Unix users that prefer to use the command-line interface, can use the `IMPORT` utility program to install the Security Runtime Files.

Once received, use the `IMPORT` utility to install the Security Runtime Files in the EasyAccess2000 `Runtime` subdirectory.

Installing Security Runtime Files (non-SSL networks)

1. Move the current directory designation the top-level EasyAccess2000 directory folder. Use the **`cd`** command (UNIX or Windows batch command).
2. Type **`import`** at the command prompt and press [**Enter**] to start the installation.
3. The `IMPORT` utility prompts you for the name of the file received from the trading partner (compressed and encrypted).
4. At the prompt, type the **file name** and **directory path** (if needed) of the received file. Press [**Enter**].
5. The `IMPORT` utility prompts you for the directory folder where you want to store the Security Runtime Files.

CAUTION These files must be installed in the EasyAccess2000 `RUNTIME` subdirectory shown in Figure 1.

6. Type **runtime** and press **[Enter]** to install the runtime files into the EasyAccess2000 `Runtime` subdirectory.
7. The `IMPORT` utility prompts for the approval code. This is a 16-character value provided by the hub-trading partner that protects the Security Runtime Files from unauthorized access. If you do not know your approval code, then contact the hub-trading partner.

8. Type the 16-character approval code and press [**Enter**].

9. The `IMPORT` utility prompts for the file directory where the `GENKEYS` utility created the `private.key` file. This file is in the EasyAccess2000 `Security` subdirectory.
10. Type the word **security** and press [**Enter**]. You have now installed the Security Runtime Files.

You can avoid the four prompts described in the steps above by providing all the information on the command-line interface. An example `import` command with the four prompts would look like:

```
>import export.rtm runtime key=0123456789ABCDEF privkey=security
```

Installation of the runtime files completes the security configuration. Additional customization of stored transfers may be necessary to configure proper network parameters. The hub-trading partner provides these specific instructions. Contact bTrade.com Product Support if this EasyAccess2000 Configuration Guide does not provide the information needed to set-up your data transfers.

Unless you need to install Security Runtime Files for a network that uses Secure Socket Layer (SSL), you can continue with section three, EasyAccess2000 *Keyword Reference*, and begin working with commands to send or receive data files.

Procedure E: Installing Security Runtime Files (IGN-I/E SSL Networks)

After ordering the SSL Internet connectivity from IBM or AT&T, the user receives a letter and a diskette containing the key pairs and password for loading the keys. If diskette has not been received, call 800-655-8865 to receive the certificates. Once the above information has been received, use the `PARSEPFX` utility to install the certificate files in the EasyAccess2000 subdirectory folder.

Installing Security Runtime Files (IGN-I/E SSL networks)

1. Move the current directory designation the top-level EasyAccess2000 folder. Use the **`cd`** command (UNIX or Windows batch command).
2. Type **parsepfx** and press [**Enter**].
3. The `PARSEPFX` utility prompts you for the directory path and the *Prime File Transfer* (PFX) file name from *IBM Global Network* (IGN).
4. Type the directory path and file name of the key file received from IGN. press [**Enter**].
5. The `PARSEPFX` utility prompts you for the PFX file password.
6. Type the **`password`** received from IGN. You have now installed the Security Runtime Files.

AS/400 Systems

To use EasyAccess2000 for data transfers on an AS/400 machine, you must have:

- Installed operating system OS/400 version V3R7M0 or above
- Installed and configured the TCP network component of AS/400
- Established a physical connection to the network to access your trading partner’s server For example, via the Internet)

EasyAccess2000 does not provide phone dialing or other functions for establishing the physical connection.

Overview

To configure and use EasyAccess2000 for the OS/400 operating system, you need to:

1. Install the EasyAccess2000 software
2. Generate the EasyAccess2000 Encryption Keys
3. Send the certificate request file to the hub-trading partner
4. Receive the Security Runtime Files from the hub-trading partner
5. Install the Security Runtime Files (Non-SSL Networks)
   or
   Install the Security Runtime Files (IGN-I/E SSL Networks)
6. Exchange data with trading partners

Configuration Files

On the OS/400 operating system there are two EasyAccess2000 configuration files (`EASYACC` and `EXFER`) provided by the hub or VAN. These files are customized prior to distribution and are installed in the `EasyAccess` library. The files are physical text files and can be edited by users.

Caution: Do not change the contents of these AS/4000 EasyAccess2000 configuration files (`EASYACC` and `EXFER`) unless specifically instructed by bTrade.com Product Support personnel.

EasyAccess2000 creates temporary work files during operation to store temporary data, such as compressed files and server file lists. These files are created in the library designated as `*CURLIB`. Use the `CHGCURLIB` command to establish the `EasyAccess` library as the current library prior to running any EasyAccess2000 file transfers.

Procedure A: Installing EasyAccess2000 AS/400 Software

EasyAccess2000 is distributed as a `save` library in `SAVEFILE` format. The library name is `EA2KLIB`.

Installing EasyAccess2000 AS/400 Software

1. Create an empty save file on the AS/400.
   An example command to type would be - **`CRTSAVF SAVEFILE`**
2. On a Windows 95/98/2000/NT system, decompress the distributed files by running the self-extracting executable file (file extension `.EXE`).
3. Upload the distributed `SAVE` file from the Windows PC to the new AS/400 `SAVE` file using a binary mode FTP transfer. An example FTP session is illustrated in this figure. Bold font indicates commands typed by the user. Comments are shown with the ← designation.

Figure 2: Example FTP Session to Transfer EaysAccess2000 Software to AS/400

```
> ftp 180.138.16.2 (IP address)            ← connect to AS/400 at network address

220 User (as/400:(none)): userid           ← type userID necessary

331 Enter password.                        ← type password needed

230 USERID logged on.

ftp> bin                                   ← switch to binary transfer mode

220 Representation type is binary IMAGE.   ← confirmation

ftp> put ea2klib.savf SAVEFILE             ← transfer installation save file

200 PORT subcommand request successful.

150 Sending file to member SAVEFILE in file SAVEFILE.

250 File transfer completed successfully.

ftp> quit                                  ← end FTP session
```

4. Use the `RSTLIB` command to unload the EasyAccess library. An example command is
   `RSTLIB SAVLIB(EA2KLIB) DEV(*SAVF) SAVF(SAVEFILE)`

Complete the installation by generating the EasyAccess2000 encryption keys and installing the security run-time files.

Procedure B: Generating EasyAccess2000 Encryption Keys

Before EasyAccess2000 can transmit secure data, you must generate encryption keys and have them certified by the hub-trading partner or import them if they are provided by your VAN. If you need to generate keys, the `GENKEYS` utility creates the encryption keys. It reads the `EASYACC` configuration file and creates two output files, `CERTREQ` and `PRIVKEY`. Follow these steps to generate the EasyAccess2000 encryption keys. If the keys have been generated by the hub-trading partner and received on diskette, skip the steps below and go to "*Procedure F: Installing Security Runtime Files (IGN-I/E SSL Networks).*"

Generating encryption keys

1. Make the EasyAccess2000 library the current library by typing the command:
   `CHGCURLIB EA2KLIB`
2. Use the `GENKEYS` utility to create the encryption keys by typing the command:
   `CALL GENKEYS`
3. When the `GENKEYS` utility prompts for random input data, type several lines of random characters to create encryption keys that are difficult to unscramble.
4. Press [**Enter**] on a blank line to complete the random entry. The `GENKEYS` utility writes your `privkey` and `certreq` files to the EasyAccess library.

The `PRIVKEY` file created by `GENKEYS` utility is a permanent key file and must be retained.

Procedure C: Sending a Certificate Request

The `CERTREQ` file contains the portion of the key that must be certified by the hub-trading partner. This file must be transmitted to the hub. The hub issues the Security Runtime Files required to transmit secure data. Most users can transmit the `CERTREQ` file to the hub by executing EasyAccess2000 and using the `SEND CERT REQ` stored transfer. Specific instructions for sending the `CERTREQ` file are provided by the hub-trading partner.

Procedure D: Receiving Security Runtime Files from Trading Partner

To complete the security configuration, you must install the Security Runtime Files generated by the hub-trading partner. The Security Runtime Files are distributed from the hub as a single compressed and encrypted file. Most users can receive the compressed file by running the EasyAccess2000 application and using the `RECEIVE RUNTIMES` stored transfer. Specific instructions for receiving the compressed file are provided by the hub-trading partner.

Procedure E: Installing Security Runtime Files (non-SSL Networks)

Once received, use the `IMPORT` utility program to install the run-time files in the `EasyAccess` library.

Installing Security Runtime Files (non-SSL networks)

1. Make the EasyAccess2000 library the current library by typing the command:
   `CHGCURLIB EA2KLIB`
2. Use the `IMPORT` utility to install the Security Runtime Files by typing the command:
   `CALL IMPORT`
3. The `IMPORT` utility prompts for the name of the compressed file received from the hub.
   Locate the file using the `LIBLIST` command
   or
   Type the file name using the `LIBRARY/FILENAME` format
   or
   Only type the file name.
4. The `IMPORT` utility prompts for the name of the library where to install the Security Runtime Files. Type the name of the EasyAccess2000 library.
5. The `IMPORT` utility prompts for the approval code. This is a 16-character value provided by the

hub-trading partner that protects the Security Runtime Files from unauthorized access. If you do not know your approval code, then contact the hub-trading partner.

6. The `IMPORT` utility prompts library where the `GENKEYS` utility created the `PRIVKEY` file. Type the name of the EasyAccess2000 library.

You can avoid the four prompts described in the steps above, by providing all the information on the command-line interface. An example `IMPORT` command with the four prompts would look like:

```
CALL IMPORT PARM('RTMFILE' 'EA2KLIB' 'KEY=0123456789ABCDEF' 'PRIVKEY=EA2KLIB')
```

Installation of the runtime files completes the security configuration. Additional customization of stored transfers may be necessary to configure proper network parameters. The hub-trading partner provides these specific instructions. Contact bTrade.com Product Support if this *EasyAccess2000 Configuration Guide* does not provide the information needed to set-up your data transfers.

Unless you need to install Security Runtime Files for a network that uses Secure Socket Layer (SSL), you can continue with *Procedure G: Exchanging Secure Data*, and begin working with commands to send or receive data files. Installation of the security run-time files completes the security configuration. If further customization of stored transfers is required, the hub-trading partner provides specific instructions.

Procedure F: Installing Security Runtime Files (IGN-I/E SSL Networks)

To complete the security configuration, install the Security Runtime Files generated and shipped by the hub-trading partner. The `KEYS`, which are distributed from the hub as a compressed and encrypted file, arrive on diskette and include a file passcode. This file must to be transferred to a physical file on the AS/400 using an FTP program (binary format). Once the keys are received into the physical file, use the `PARSEPFX` utility to generate the `Publkeys` and `Privkeys` files.

The `Publkeys` and `Privkeys` files are Comm-Press200 Security Runtime Files:

- The `Publkeys file` contains the trading partners' public RSA certificates used to *encrypt* and verify digital-signed data.
- The `Privkeys` file contains the EasyAccess2000 user's private RSA key used to decrypt and digital-signed data.

Installing Security Runtime Files (IGN-I/E SSL networks)

1. Make the EasyAccess2000 library the current library by typing the command,
   `CHGCURLIB EA2KLIB`
2. Use the `PARSEPFX` utility to install the Security Runtime Files by typing the command,
   `CALL PARSEPFX`
3. The `PARSEPFX` utility prompts for the name of the compressed file received from the hub.
   Locate the file using the `LIBLIST` command
   or
   Type the file name using the `LIBRARY/FILENAME` format
   or
   Only type the file name.
4. The `PARSEPFX` utility prompts for the name of the library where the *Prime File Transfer* (PFX) file is installed. Type the name of the EasyAccess2000 library.

5. The `PARSEPFX` utility prompts for the PFX password—an eight-character string provided by the hub-trading partner and used to protect the PFX file from unauthorized access. If you do not know the PFX password, contact the hub-trading partner.
6. The `PARSEPFX` utility displays the message: “`Key Import Successful`” when the Security Runtime Files are successfully installed.

You can avoid the three prompts described in the steps above, by providing all the information on the command-line interface `CALL`. An example `PARSEPFX` command with the three prompts would look like:

```
CALL PARSEPFX PARM('PFXFILE'  'EA2KLIB'  'KEY=012345678')
```

Installation of the PFX files completes the security configuration. Further customization of stored transfers may be required. The hub-trading partner provides specific instruction.

Procedure G: Exchanging Secure Data

You can exchange secure data with EasyAccess2000 by calling it from the AS/400 command line or from the command-line interface application. Use the `PARM` keyword on the `CALL` statement to specify the names of stored transfers, login User ID and password, and any other required information. This information may also be provided via a command file. If so, then specify the name of the command file in the `PARM` keyword.

Exchanging Secure Data using AS/400 EasyAccess2000

1. Use the AS/400 command line
   or
   Use the EasyAccess2000 command-line interface application with the name `EA2K400C`.
2. Use the `PARM` keyword on the `CALL` statement to specify this information:
   Name of a stored transfer,
   User ID and password required to logon to hub-trading partner server,
   Compression and decompression program options,
   or
   Name of the command file that contains this information.

When using the command file option from the command-line interface, the command file must be in a physical file format and should contain the appropriate transfer, compression, and decompression parameters.

The next table displays several command syntax examples used to implement an EasyAccess2000 data transfers on an AS/400 computer.

Table 9: AS/400 Data Transfers - Command Syntax Examples

Example	Command Example
Command file name	`CALL EA2K400C PARM(‘TRANSFER=CMDFILE=LIBRARY/FILE’)`
Use a stored transfer called from command-line interface	`CALL EA2K400C PARM(‘TRANSFER=TRANSFER’)`
Use a stored transfer with a blank in its name.	`CALL EA2K400C PARM(‘TRANSFER=”TEST TRANSFER”’)` Use double quotes to surround the stored transfer name.
Creating a stored transfer for later use	`CALL EA2K400C PARM(‘TRANSFER=NAME=”NEW TRANSFER” ‘PASSWD=password’ ‘NETWORK=network’ ‘ASCII’ ‘CRLF’ ‘COMPRESS’ ‘SECURE’)`

The new transfer name and parameters are appended to the EXFER file and can be called later by using only the stored transfer name. Specifying the transfer and compression options later will not be necessary.

AS/400 Operating System-specific EasyAccess2000 Considerations

To simplify your command syntax when running utility programs or EasyAccess2000 data transfers, you may want to use the CHGCURLIB (change current library) command to make the EasyAccess2000 library the current library

Security Runtime Files

The Security Runtime Files must be available when transmitting secure data with EasyAccess2000. Be sure the EasyAccess2000 library, or the library where IMPORT utility has installed the Security Runtime Files, is available via the LIBLIST command.

Temporary Work Files

EasyAccess2000 creates several temporary files as part of its normal application processing. These files are written to the AS/4000 designated “current library”. This is another reason for making the EasyAccess library the current library prior to EeasyAccess2000 execution (with the `CHGCURLIB` command).

During transmission, EasyAccess2000 creates temporary files named **`SYSUT1`** and **`SYSUT2`** in the current directory. These files hold directory listings and copies of compressed and secured data files. System defaults are usually adequate for creating these temporary files; however, if you send or receive large files you may need to pre-create one or both of the temporary files with an adequate size to hold the data. If this is the case, then create these files as physical files with a record length of 256 bytes. You may need to experiment with the number and size of the record extents to allocate files of the desired sizes.

Naming and Allocating Work Files

Specify the file names used to send and receive data by using the `LIBRARY/FILENAME(MEMBER)` syntax. If the file is available via the **`LIBLIST`** command, then you can omit the `LIBRARY` portion of the command. If the first (or only) library member is needed, then you can omit the `(MEMBER)` portion of the command.

When receiving data, EasyAccess2000 creates the output files if they do not exist. However, the files are created in the current library with default values for maximum record length and file size. If the defaults are not acceptable, then you should create the files with the appropriate with number and size of the record extents, prior to receiving the transmitted data.

MVS Systems

To use EasyAccess2000 for data transfers on an MVS OS/390 machine, you must have:

- Installed operating system MVS version 2.6 or higher
- Included the MVS feature of Language Environment version 1.9
- Included the appropriate C++ language support feature
- Installed and configured TCP/IP version 3.4 or higher
- Established a physical connection to the network to access your trading partner’s server
  For example, via the Internet)

EasyAccess2000 does not provide phone dialing or other functions for establishing the physical connection.

Overview

To configure EasyAccess2000 for the MVS operating systems you need to:

1. Install the EasyAccess2000 software using a diskette, CD-ROM, or distribution tape.
2. Generate the EasyAccess2000 Encryption Keys
3. Send the certificate request file to the hub-trading partner.
4. Receive the Security Runtime Files from the hub-trading partner.
5. Install the Security Runtime Files.
6. Exchange data with other trading partners.

MVS Libraries

EasyAccess2000 is distributed as a self-extracting, compressed installation package. Run the self-extracting file on a Windows 95/98/NT/2000 system to decompress the MVS libraries. The MVS libraries are created using the `TRANSMIT` command on TSO. The MVS libraries are named:

- `EA2KMVSC.LIB` **-** EasyAccess2000 application and utilities load library
- `EA2KMVSC.CTL` **-** Sample *Job Control Language* (JCL) and configuration data files

Procedure A: (Disk/CD-ROM Option) Install EasyAccess2000

The libraries were constructed using the MVS *Time Sharing Option* (TSO) `Transmit` command The disk files are compressed and self-extracting. Once the files are transmitted to the MVS system, you will use the TSO `Receive` command to unload the files into a *partitioned data set* (PDS).

Install EasyAccess2000 (Disk/CD-ROM Option)

1. Move the installation package to a Windows 95/98/NT/2000 file system.
2. Double-click to start the installation program.
3. To install the files on MVS, begin by uploading the files using a PC-to-mainframe file transfer program
   or
   Use a DOS file transfer program (FTP) to upload the files.

For MVS operating system, the files must use the 80-byte, fixed-record format. If you use the DOS FTP program to upload files, you must pre-allocate space for the `EA2KMVSC.LIB` and `EA2KMVSC.CTL` libraries using JCL specifications similar to this:

```
EALIB.FILE DD DSN=USER.EALIB.FILE, DISP=(NEW,CATLG),
                 UNIT=SYSDA, SPACE(TRK,5,5)),
                 LRECL=80,BLKSIZE=3120, RECFM=FB)
```

4. Use DOS FTP to transfer the files into the existing datasets. Send the files as binary files (for example, no ASCII/EBCDIC translation and no carriage return/line feed processing).

An example FTP session is illustrated in this figure. Bold font indicates commands typed by the user. Comments are shown with the 🡐 designation.

Figure 3: Example FTP Session to Transfer EasyAccess2000 Software to MVS System

```
> ftp 180.138.16.2 (IP address)              🡐 connect to MVS/OS390 at network address
220 User (none): userid                      🡐 type userID necessary
331 Enter password.                          🡐 type password needed
230 USERID logged on.
ftp> bin                                     🡐 switch to binary transfer mode
220 Representation type is binary IMAGE.     🡐 confirmation
ftp> put ea2kmvsc.lib ‘user.ealib.file” rep       🡐 transfer first library
200 PORT subcommand request successful.
125 Storing data set user.ealib.file
250 File transfer completed successfully.
ftp> put ea2kmvsc.ctl ‘user.eactl.file” rep       🡐 transfer second library
200 PORT subcommand request successful.
125 Storing data set user.eactl.file
250 File transfer completed successfully.
ftp> quit                                    🡐 end FTP session
```

5. Once the files are on the mainframe, issue the TSO `RECEIVE` command to unload the files into a PDS with a command similar to - `RECEIVE INDA(‘USER01.EALIB.FILE’)`.

This partitioned data set (PDS) is in MVS LOADLIB format with a record format (RECFM) of U and a `BLKSIZE` of `6144` bytes. The PDS does not need to be pre-allocated because the TSO `RECEIVE` command allocates it during the unloading operation. If you wish to unload the file into an existing library, it must have the same `RECFM` and `BLKSIZE` given above.

6. The `RECEIVE` command issues a prompt before it unloads the file; respond with the name of the PDS where the Comm-Press2000 modules are installed.

Figure 4: Example TSO Receive Command - Unload EasyAccess2000 Software to MVS PDS

```
> receive
Enter restore parameters or DELETE or END +      ← prompt from RECEIVE
                                                   DA('USER01.LIBRARY')
                                                   <=== enter PDS name
```

Complete the installation by generating the EasyAccess2000 encryption keys and installing the security run-time files.

Procedure A: (Tape Option) Install EasyAccess2000

The EasyAccess2000 MVS distribution tape contains two files, the library containing the MVS load modules and the sample *job control language* (JCL) containing routines to run the `COMPRESS` and `DECOMP` programs. To install the EasyAccess2000 software directly on the MVS mainframe, we perform several steps within the context of a batch JCL job.

Install EasyAccess2000 (Tape Option)

1. Provide appropriate accounting codes and information
2. Unload the EasyAccess2000 program and utilities library (`COPY1` step) distribution tape contents (`VOL=CMTAPE`) to a temporary file on disk (`USER.LOADLIB`)
3. Unload the EasyAccess2000 sample JCL and configuration files (`COPY2` step) distribution tape contents (`VOL=CMTAPE`) to a temporary file on disk (`USER.SAMPLIB`)

Use the example MVS JCL shown in the next figure to unload the EasyAccess2000 software distribution tape contents to the MVS mainframe.

Figure 5: JCL Used to Install EasyAccess2000 on MVS from a Distribution Tape

```
//UNLOAD                JOB   (ACCOUNT INFO),'user info',CLASS=A,MSGCLASS=X
//COPY1                 EXEC  PGM=IEBCOPY
//SYSPRINT              DD    SYSOUT=*
//INDD1                 DD    DSN=CMMPRESS.LOADLIB,DISP=(OLD,PASS),
//                      UNIT=TAPE,VOL=SER=CMTAPE,LABEL=(1,SL)
//OUTDD1                DD    DSN=USER.LOADLIB,DISP=(NEW,CATLG),UNIT=SYSDA,
//                      SPACE=(CYL,(2,2,5)),BLKSIZE=6144,RECFM=U
//SYSIN                 DD    *
COPY I=INDD1,O=OUTDD1
//*
//COPY2                 EXEC  PGM=IEBCOPY
//SYSPRINT              DD    SYSOUT=*
//INDD2           DD    DSN=CMMPRESS.SAMPLIB,DISP=OLD,
//                      UNIT=TAPE,VOL=SER=CMTAPE,LABEL=(2,SL)
//OUTDD2                DD    DSN=USER.SAMPLIB,DISP=(NEW,CATLG),
//                      UNIT=SYSDA,SPACE=(TRK,(5,5,5)),BLKSIZE=3120,
//                      LRECL=80,RECFM=FB
//SYSIN                 DD    *
COPY I=INDD2,O=OUTDD2
//
```

Procedure B: Generating EasyAccess2000 Encryption Keys

If you have received encryption keys from the hub-trading partner on a diskette, running **GENKEYS** is not necessary, you can skip the steps below, and go to "*Procedure D: Installing Security Runtime Files*."

Before EasyAccess2000 can transmit secure data, you must generate encryption keys and have them certified by the hub-trading partner or your VAN. The GENKEYS utility generates the encryption keys.

Generating encryption keys

1. Locate the example GENKEYS JCL job is distributed in the EA2KMVSC.CNTL library.
2. Follow the instructions found in the example GENKEYS JCL job.
3. Revise the example GENKEYS JCL job by following the documented instructions.
4. GENKEYS reads the EASYACC configuration file and creates your privkey and certreq files.

CAUTION The PRIVKEY file created by GENKEYS utility is a permanent key file and must be retained.

Procedure C: Sending a Certificate Request

The CERTREQ file contains the portion of the key that must be certified by the hub-trading partner. This file must be transmitted to the hub. The hub will issue the security runtime files required to transmit secure data. Most users can transmit the CERTREQ file to the hub by executing EasyAccess2000 and running the SEND CERT REQ stored transfer. The hub-trading partner provides specific instructions for sending the CERTREQ file.

Procedure D: Installing the Security Runtime Files

To complete the security configuration, you must install the Security Runtime Files generated by the hub-trading partner or your VAN. The Security Runtime Files are distributed from the hub/VAN as a compressed and encrypted file. Most users can receive the compressed file by executing EasyAccess2000 and running the RECEIVE RUNTIMES stored transfer. Specific instructions for receiving the compressed file are provided by the hub-trading partner.

Once received, use the IMPORT (or in some cases, the CMDPARSE) utility to install the run-time files. Sample IMPORT and CMDPARSE JCL jobs are distributed in the EA2KMVSC.CNTL library. Instructions for running the job and installing your security run-time files are included in the sample JCL. Use the two utilities:

- Import for working with non-SSL networks
- CMDPARSE for working with the IGN-I/E SSL networks

Installation of the security runtime files completes the security configuration. Further customization of stored transfers may be required. The hub-trading partner provides specific instructions.

Procedure E: Exchanging Secure Data

Exchange secure data with EasyAccess2000 by running a JCL batch job or a command list (CLIST) using TSO. An example JCL batch job and CLIST are distributed in the EA2KMVSC.CNTL library. Instructions for modifying the JCL and CLIST, plus running EasyAccess2000 JCL, are included in the examples.

To define the data files to exchange:

- Use the actual data set names
- Include JCL statements (specifically DD statements) to reference your files
  Use the syntax DD:DDNAME (where 'DDNAME' is the name you coded on the actual DD statement) to refer to files by their DD statements.

Figure 6: Example DD Statement for EasyAccess2000 Data Transfer Files

```
//RECVFILE DD DSN=RECEIVE.FILE,DISP=(NEW,CATLG),
//            UNIT=SYSDA, SPACE=(TRK,(5,5)),
//            LRECL=80,BLKSIZE=0,RECFM=FB
```

EasyAccess2000 Keyword Reference

Command-line Interface Keywords

CAUTION

Requirement: The EasyAccess2000 command file supports its own `CMDFILE=` keywords *or* the `IEBASE` utility keywords. You can use one set of keywords or the other, but not both within a single EasyAccess2000 data transfer. In Table 10, you can use the `CMDFILE=` or the `IEBASE` keyword, but not both.

These keywords can be used on the actual EasyAccess2000 command-line interface:

Table 10: EasyAccess2000 Command-line Interface Keywords

Keyword	Description and Usage of Keyword
`CMDFILE=`	Names the file containing EasyAccess2000 keywords to use. The command file completely controls the data transfer. The following tables in this chapter provide: o Summary of supported command file keywords o Building rules used with a command file o Examples of command files If no command file is specified (using the `CMDFILE=` keyword) when you run the EasyAccess2000 command-line interface, the program uses data transfer instructions previously setup using the graphical user interface (GUI) or manually with a text editor. The actual instructions and values are stored in the `easyacc.ini` and `exfer.ini` files.
`HELP`	Displays on-line usage guidelines. This keyword requires no value.
`IEBASE`	Performs IEBASE functionality. Program reads, parses an "IBM EXPEDITE-style" file—`basein.msg`, and creates data transfers to be run. This keyword requires no value.
`INIPATH=`	Overrides the directory folder that defines the "*root directory*" for the installed EasyAccess2000 software. This is the directory contains the `easyacc.ini` and `exfer.ini` files as well as the sub-folder directories required (for example, `security`, `runtime`, `temp`, and so on). If the INIPATH= keyword is not specified, the program uses the **current working directory** as the EasyAccess2000 "*root directory*".
`MODE=`	`BATCH` or `GUI`. Applies to the GUI version of EasyAccess2000 only. Specifies that the GUI program is to run in command-line interface mode. `MODE=GUI` is the default, ordering the EasyAccess2000 application to run in GUI mode.
`VALIDATE_TRANSFERS_ONLY`	Validate the specified data transfers *only*, report on their validity, and exit.
`RESET`	Ignore any previously failed data transfers, which would otherwise attempt to restart.

Keyword	Description and Usage of Keyword
`GENKEYS`	Generate a public/private key pair, bundle the pair into a certificate request, plus create and run a transfer to send the request to a location designated in the `SECURITY` section of the `easyacc.ini` file.
`RECEIVE_RUNTIMES`	Create and run a data transfer to receive your Security Runtime Files from a location designated in the `SECURITY` section of the `easyacc.ini` file. The Security Runtime Files are automatically installed, giving EasyAccess2000 access to the public keys of your trading partners.

Command File Building Rules

As you construct a keyword command file (see remaining tables in this section of the *EasyAccess2000 Customization Guide*), keep these rules in mind.

Table 11: Command File Building Rules

Rule	Command File Building Rule Description
1	Text comments to the right of any pound ('#') character are ignored. **Exception:** pound ('#') character occurs within text that enclosed by single or double quotes.
2	Blank lines in the command file are ignored. Use this to create white space, so you can clearly see the functionality.
3	Command-line interface and command file parameters can be delimited using: o Single quotes - ' o Double quotes - " o Parentheses - `()` o Square braces - `[ ]` o Curly braces – `{ }` Sub-expressions (expressions within expressions) can be delimited within main expressions by using a different delimiter shown above. **Example:** `TRANFER=(name='my trans'… OTHER_COMP_PARMS='parm1 parm2' …)`
4	Spaces are used only to separate keyword/value pairs and are otherwise ignored unless they are within a delimited expression.: **Example:** `TRANSFER=  "My transfer"` is the same as `TRANSFER="My Transfer"` **Example:** `TRANSFER  =  "My Transfer"` is illegal. (TRANSFER= is the keyword, that means no spaces allowed within the keyword itself) **Example:** `TRANSFER=  My Transfer` is illegal (it is saying to use transfer name "My", not "My Transfer") because there are no quotes around the transfer name.
5	The end-of-line has no special significance. You can put all your keywords on one line or spread them out across multiple lines within the command file. See additional examples throughout the next major section of the *EasyAccess2000 Configuration Guide*.

Rule	Command File Building Rule Description
6	The keywords are not character case-sensitive. The keyword values may be case-sensitive, depending on the server's operating system on the other end of the data transfer. **Client Example:** `transfer=` and `Transfer=` are the same as `TRANSFER=` **Server Example:** `LOGINUSERID=john_smith` may or may not be equal to `LOGINUSERID=John_Smith` (depending upon server's operating system)
7	Up to 25 transfers can be *created* within a command file. Each transfer created is added to the list of transfers to be run.
8	Up to 25 *existing* transfers can be specified in the command file. Each specified existing transfer is added to the list of transfers to be run.
9	An unlimited number of trading partners can be added to your Trading Partner Address book using the `TPBOOK=` command keyword.
10	Keywords do not contain spaces (blanks) within the names, only underscore characters (_).

Supported Command File Keywords

The following keywords can be used in a command file that is specified with the `CMDFILE=` command-line interface keyword as the EasyAccess2000 program is being invoked.

Table 12: EasyAccess2000 Command File Keywords

Keyword	Description and Usage of Command File Keyword
`GENKEYS`	Generates a public/private key pair, bundles the pair into a certificate request, and finally creates and runs a stored transfer that sends the certificate request to a location designated in the `SECURITY` section of the `easyacc.ini` file. (See *`easyacc.ini` File Reference* section).
`PASSLOC=`	Specifies the passphrase file location for encrypting the keys. Provides user with the ability to specify the location of a security token. If not specified, then a default value is used.
`RECEIVE_RUNTIMES`	Create and run a transfer that receives the Security Runtime Files from a location designated in the `SECURITY` section of the `easyacc.ini` file.
`QUERY_LIST`	Create and run a transfer that receives a list of available files from the server. The results from this keyword command, is the `list.fil` file stored the EasyAccess2000 `Temp` subdirectory folder (see Figure 1.)
`QUERY_FILE=`	Specifies the fully qualified file name used to receive the server file list transmitted when the `QUERY_LIST` keyword is used. If this keyword is absent, the file list is written to the default file name (`list.fil`) in the EasyAccess2000 `Temp` subdirectory folder (see Figure 1.)
`RECEIVE_AUDIT_LOGS`	Create and run a transfer to receive an audit report from the current server of files sent and received. Creates as output the `audit.log` file in the EasyAccess2000 `Temp` subdirectory.
`AUDIT_FILE=`	Specifies the fully qualified file name used to receive the audit logs transmitted when the `RECEIVE_AUDIT_LOGS` keyword is used. If this keyword is absent, the audit logs written to the default file name (`audit.log`) in the EasyAccess2000 `Temp` subdirectory folder.

Keyword	Description and Usage of Command File Keyword
`AUDIT_START_DATE=`	Specify the starting date of the audit report. Used with `RECEIVE_AUDIT_LOGS` keyword to specify the starting date in the format - `yyyymmdd`
`AUDIT_END_DATE=`	Specify the ending date of the audit report. Used with `RECEIVE_AUDIT_LOGS` keyword to specify the ending date in the format - `yyyymmdd`
`CMDPARSEPFX`	Imports your private/public encryption key pair from *IBM Global Network* (IGN) for use with encryption across the IGN-Information Network networks. This keyword installs the keys for SSL networks.
`IMPORT`	Installs the Security Runtime Files giving EasyAccess2000 access to the public keys of your trading partners. This keyword is used mostly for non-SSL networks.

Network Parameter Override Keywords

The following keywords can be used at the command-line interface to override network parameters as the EasyAccess2000 program is being invoked.

Any overrides to this network's data (using the keywords described in the table) only apply for a single program run, unless the `SAVE` keyword is used.

Table 13: EasyAccess2000 Network Parameter Override Keywords

Keyword	Format	How this Keyword Overrides Network Parameters
`NETWORK=`	**Text**	Defines the current network is to be used for this communications session. The text *must* match one of the networks defined in your `easyacc.ini` file.
`IP=`	**Host Name** **IP Address**	Overrides the server address.
`IP2=`	**Host Name** **IP Address**	Overrides the backup server address.
`FTPUSERID=`	**Text - name**	Overrides the server logon user ID.

`FTPPASSWD=`	**Text**	Overrides the server logon password.
		Caution: If you use this keyword, your password is entered as clear text on the command-line interface. *You should check with your security manager before using this keyword*.
		Restriction: Requires network server that supports this functionality. Password changing syntax depends upon the network server.
		Formats:
		`FTPPASSWD=old-password/new-password or`
		`FTPPASSWD=old-password/new-password/new-password`

Keyword	Format	How this Keyword Overrides Network Parameters
NETSTYLE=	**Text**	**FTP Communication Networks:** o "GENERIC" o "GENERIC-DOS" o "GENERIC_SSL" o "IGN-IE" o "FEDEXNET" o "WALMART" o "EAFTP" o "MARK_III" (GEIS) o "EDI*Express" (GEIS) o "EDISwitch" (GEIS) o "CONNECTMAIL" o "Sterling-Commerce" (Sterling) o "MCI-Edi*Net" o "QRS-ELINK" **SMTP/POP3 Mail Servers:** o "EDI-INT" o "GISB-CLIENT" o "GISB-SERVER" **Compression and encryption without file transfer (archiving)** o "LOCAL-ARCHIVE"
CASE=	**U** or **L**	Network text case-sensitivity setting. CASE=U – convert and send data to the server in upper-case format. CASE=L – send data to the server unchanged.
Requirement: The next five keywords are applicable to FTP Communication Networks only!		
COMMAND_OVER_DATA=	**Y** or **N**	COMMAND_OVER_DATA=Y – use command-over-data variant of FTP, which uses a single socket connection. COMMAND_OVER_DATA=N – use the conventional FTP command and data socket connections.
CONTROLPORT=	**Integer**	*Overrides* the standard command-channel port number for communicating with the server
PASSIVE=	**Y** or **N**	*Overrides* the passive mode setting for the session. PASSIVE=Y is used in certain circumstances to permit data transfers through a server's firewall.
SSL=	**Y** or **N**	SSL=Y - use SSL 3.0 when establishing a session with the server. SSL=N – do not use *Secure Socket Layer* (SSL) 3.0 with server.
SITEDELAY=	**Integer** (secs)	**Server FTP command delay** - a value (seconds) to wait prior to sending each FTP command. Helps with specific timing problems. SITEDELAY=0 – Usual default for no delay. SITEDELAY=N – Wait N seconds before sending FTP command.
Use keywords to save network override keywords and permanently change network parameters.		
SAVE	none	Save the network data specified on the command-line or in the command file in the easyacc.ini file, causing the data to be permanently in effect until changed. **Default:** do not save network data and the keywords apply only for the duration of the current program run.

SAVE_ONLY	none	Acts like the SAVE keyword, except EasyAccess2000 exits (does not perform a data transfer) after saving the specified network data in the **easyacc.ini** file.

File Transfer Keywords

These keywords can be used at the command-line interface to specify the running of data transfers.

Table 14: EasyAccess2000 File Transfer Keywords

Keyword	Description and Usage of File Transfer Keyword
TRANSFER=	**Transfer Options:** o Specify an existing data transfer by name o Create a new named data transfer and add it to the list of data transfers to run **Select Existing Transfer (Format):** o TRANSFER=trans_name or o TRANSFER="trans name" Quotes are required *if* the transfer 'name' contains spaces. trans_name – actual value assigned this keyword **Create a New Data Transfer (Format):** o `TRANSFER=(NAME=value keyword=value … keyword=value)` where all keywords that define the data transfer are within the (*required*) parentheses delimiters.
NAME=	Name of the new data transfer. Required keyword when the create new data transfer format is used `TRANSFER=(NAME=value keyword=value … ).`
LOGINUSERID=	Specifies a server logon user ID other than the default network logon ID used prior to executing the data transfer. **Requirement:** Used with the LOGINPASSWD= keyword.

LOGINPASSWD=	Specifies a server logon password other than the default network logon password used prior to executing the data transfer.
	Restriction: Used with the LOGINUSERID= keyword.
	Caution: If you use this keyword, your password is entered as clear text on the command-line interface. *You should check with your security manager before using this keyword*.
	Requirement: Requires network server that supports this functionality. Password changing syntax depends upon the network server.
	Formats:
	`LOGINPASSWD=old-password/new-password or`
	`LOGINPASSWD=old-password/new-password/new-password`

Keyword	Description and Usage of File Transfer Keyword
PROXY_TYPE=	PROXY_TYPE=1, specifies a Proxy Server is to be used to connect to the target FTP server and the Proxy Server requires a login User Id and Password. Sequence of Events: • A connection is first established using the IP address specified by, the IP keyword (primary IP address or domain name) • The Proxy Server login takes place using the User Id and password specified by keywords PROXY_USERID and PROXY_PASSWD • The target FTP server login occurs using the user ID and password specified by the FTPUSERID and FTPPASSWD keywords.
PROXY_USERID=	For PROXY_TYPE=1, specifies the login user Id for the Proxy Server
PROXY_PASSWD=	For PROXY_TYPE=1, specifies the login Password for the Proxy Server
These keywords control sending a data transfer to a remote server. **Requirement:** You can use the SEND= keyword or the SENDEDI= keyword, *but not both*.	
SEND=	Specifies the fully qualified file name of a file to be sent to the server.
SENDEDI=	Specifies the fully qualified *EDI* file name to be sent to the server.
SENDUSERID=	Name of User Id (mailbox) on the server to receive the file being sent. For *IBM Global Networks* (IGN), if the SENDEDI= keyword is specified, the SENDUSERID= keyword specifies the *Alias Table* to be used with the data transfer.
SENDCLASS=	Specifies the Class or *Application Reference Field* (APRF) to receive the file being sent to the remote sever. A set of classes that an EDI application can receive data. EasyAccess2000 uses these classes to filter EDI data during stored transfers.
SENDAPRF=	Same as SENDCLASS= keyword.
TO_ARCHIVE=	NETSTYLE='Sterling-Commerce' Specifies the directory folder and file name to receive the secured file. **Requirement:** Dataguard product only
These keywords control receiving a data transfer from remote server.	
RECEIVE=	Specifies the directory path and file name that receives the data transfer of mailbox entry downloaded from the server.
RECEIVEEDI=	Specifies the directory path and *EDI* file name that receives the data transfer of mailbox entry downloaded from the server.
RECEIVEUSERID=	Specifies that only files sent to your mailbox from this User ID are to be downloaded.
RECEIVECLASS=	Only files with this specified Class or *Application Reference Field* (APRF) can be downloaded and received from the remote sever. EasyAccess2000 uses these classes to filter EDI data during stored transfers.
RECEIVEAPRF=	Same as RECEIVECLASS= keyword.
FROM_ARCHIVE=	NETSTYLE='Sterling-Commerce' Specifies the directory folder and file name of the secured file to be accessed. (Dataguard product only)

EasyAccess2000 Send Data Transfer Override Keywords

These keywords can be used to override the Comm-Press2000 encryption and decryption default parameters specified for this “send” data transfer.

Table 15: EasyAccess2000 Send Data Transfer Override Keywords

Keyword	Keyword Values	Usage of Keyword to Override the Comm-Press2000 Parameter
COMPRESS=	**Y** or **N**	COMPRESS=Y, compress a file before it is sent to the server.
SECURE=	**Y** or **N**	SECURE=Y, encrypt a file before it is sent to the server.
FILTER=	**Y** or **N**	FILTER=Y, invokes the filter algorithm described in *request for comment* (RFC)-1113 to convert the compressed data from binary into text format. Filtered data is always transmitted as a text file. Use FILTER= keyword when the: o Data communication environment or security policy does not permit transparent data transmission o Sent data contains a combination of compressed and uncompressed data o EDI, PF, or SECFILE keyword options are used o Data is transmitted between different operating systems **Example:** EDI data on a PC that is sent to an AS/400, should be compressed with the EDI and FILTER= keywords. The compressed file is sent to the AS/400 as text.
SENDASCII=	**Y** or **N**	SENDASCII=Y, translates the data to ASCII or EBCDIC (if necessary) depending on the computer operating system where the data is decompressed. The ASCII= and CRLF= options should always be used when compressing text files.
CRLF=	**Y** or **N**	This option causes EasyAccess2000 to convert delimiter characters (for example, line feeds or carriage return/line feed pairs) into record separators. On AS/400 and MVS machines, EasyAccess2000 inserts record separators at the end of each input record. During decompression, the delimiter characters that are appropriate for the target platform replace the record separators. On PC and UNIX workstations, this keyword forces an x'1A' character to be treated as an end-of-file marker unless the IGNORE1A= option is also chosen in Comm-Press2000. The ASCII and CRLF options should always be used when compressing text files.
DELETE_AFTER_SEND=	**Y** or **N**	DELETE_AFTER_SEND=Y, delete the file after it has been successfully sent.
PERPETUAL_SEND=	**Y** or **N**	PERPETUAL_SEND=Y, make the transfer repeat its send-cycle as specified by the RETRY=, MAX_RETRY=, and RETRY_DELAY= keywords.

`OTHER_COMP_PARMS=`	**Text**	Specifies advanced Comm-Press2000 compression parameters. Refer to the *Comm-Press2000 User Guide* for supported keywords. The advanced parameters are typed just as they would appear on the command-line interface of Comm-Press2000. **Example:** `"TRANSFER=(NAME=mytransfer … OTHER_COMP_PARMS='lrecl=72 delim=250')"` **Note:** The advanced parameters can be delimited using single or double quotes, parentheses, or square or curly braces.

EasyAccess2000 Receive Data Transfer Override Keywords

The following keywords can be used to override the Comm-Press2000 encryption and decryption default parameters specified for this “receive” data transfer.

CAUTION

Requirement: You can use the `APPEND=` or the `AUTOEXT=` keyword in a receive data transfer, *but not both*.

Table 16: EasyAccess2000 Receive Data Transfer Override Keywords

Keyword	Keyword Values	Usage of Keyword to Override the Comm-Press2000 Parameter
`APPEND=`	**Y** or **N**	`APPEND=Y`, append all received downloaded files into the file specified by the `RECEIVE=` or `RECEIVEEDI=` keyword. As each new data transfer is received, the data is appended to the file
`AUTOEXT=`	**Y** or **N**	`AUTOEXT=Y`, "auto-extent" (create a unique file extension as each date is received) the file name specified by the `RECEIVE=` or `RECEIVEEDI=` keyword. As each file is received, it is given a file name with a unique numeric extension (that is, 001, 002).
`RECEIVEASCII=`	**Y** or **N**	`RECEIVEASCII=Y`, treat the file being downloaded from the server as an ASCII file.
`UNCOMP=`	**Y** or **N**	`UNCOMP=Y`. If the received files contain valid, uncompressed data, along with compressed data, Comm-Press2000 copies the uncompressed data to the output files as it decompresses. If `UNCOMP=N`, then any data that occurs between the compressed segment end and the beginning of the next compressed segment is assumed to be “pad” characters and ignored. This option is not valid in combination with the EDI parameter, since all data outside the EDI envelope is ignored and passed “as-is” by default.
`PERPETUAL_RECEIVE=`	**Y** or **N**	`PERPETUAL_RECEIVE=Y`, make the transfer repeat its receive-cycle as specified by the `RETRY=`, `MAX_RETRY=`, and `RETRY_DELAY=` keywords.

`OTHER_DECOMP_PARMS=`	**Y** or **N**	Specifies advanced Comm-Press2000 decompression parameters. Refer to the *Comm-Press2000 User Guide* for supported keywords. **Requirement:** The advanced parameters are typed as they appear on the command-line interface of Comm-Press2000. **Example:** `"TRANSFER=(NAME=mytransfer … OTHER_DECOMP_PARMS='unwrap delim=250')"` **Note:** The advanced parameters can be delimited using single or double quotes, parentheses, or square or curly braces.

Send and Receive Pre-processing/Post-processing

These keywords define send and receive pre-/post-processing options while you are creating a transfer. That is, EasyAccess2000 runs a program before or after you have it send or receive data transfers. **Requirement:** You must use these keywords inside a transfer definition.

Processing Keywords

Table 17: Pre-processing and Post-processing Keywords

Keyword	Usage in EasyAccess2000 Commands
`SEND_VERIFY=`	`SEND_VERIFY=Y`, check for the existence of the file(s) used in the created stored transfer. The default is to check that the file(s) exists, to catch typing errors. However, if you are executing a Send Pre-processing program, which creates the files to be sent, then you want to disable the checking (`SEND_VERIFY=N`), since the files may not exist until the transfer is run.
`RECEIVE_VERIFY=`	**Requirement:** Used for Dataguard product only. `RECEIVE_VERIFY=Y`, check for the existence of the file(s) you are telling it to unsecured at the time the transfer is being created. The default is to check that the file(s) exist, to catch typing errors. However, if you are executing an unsecured (Receive) pre-processing program that creates the files to be unsecured (`RECEIVE_VERIFY=N`), then you want to disable the checking, since the files may not exist until the transfer is run.
These four keywords use the Program Selection and Outcome Keywords found in the next table to select a processing program/script/command and test its return code. Examples of the command syntax and keyword usage are shown to illustrate how to construct these commands on different operating systems.	
`PRE_SEND=`	Specifies a processing program/script/command to run *before* the *send-cycle* of a data transfer.
`POST_SEND=`	Specifies a processing program/script/command to run *after* to the *send-cycle* of a data transfer.
`PRE_RECEIVE=`	Specifies a processing program/script/command to run *before* the *receive-cycle* of a data transfer.
`POST_RECEIVE=`	Specifies a processing program/script/command to run *after* to the *receive-cycle* of a data transfer.

Program Selection and Outcome Keywords

The PRE_SEND=, POST_SEND=, PRE_RECEIVE=, and POST_RECEIVE= keywords use the following syntax and keywords to specify program/script/command or command-line file to be run. The keywords also describe the conditional testing done to check if the program/script/command ran successfully.

Table 18: Program Selection and Outcome Keywords

Keyword	Keyword Usage in the Pre-processing and Post-processing Keywords
CMDLINE=	Specifies the directory path and the file name of the program, batch file, script file, or operating system command to be run. Program arguments are passed to the program/script/command within the quotes used with this keyword. (See command syntax examples in the next section.)
RETCODE=	RETCODE=Integer. Specifies a return code value to be used in determining if the program/script/command runs successfully or not.
The next eight keywords supply a conditional test of the CMDLINE= program/script/command's *actual* return code and the specified RETCODE= return code number.	
SUCCEEDS_IF_GT	The program/script/command *succeeded* if the return code is *greater than* the value specified by the RETCODE= keyword.
SUCCEEDS_IF_LT	The program/script/command *succeeded* if the return code is *less than* the value specified by the RETCODE= keyword.
SUCCEEDS_IF_EQ	The program/script/command *succeeded* if the return code *equals* the value specified by the RETCODE= keyword.
SUCCEEDS_ALWAYS	The program/script/command *always succeeds,* regardless of its return code value.
FAILS_IF_GT	The program/script/command *failed* if the return code is *greater than* the value specified by the RETCODE= keyword.
FAILS_IF_LT	The program/script/command *succeeded* if the return code is *less than* the value specified by the RETCODE= keyword.
FAILS_IF_EQ	The program/script/command *failed* if the return code *equals* the value specified by the RETCODE= keyword.
FAILS_ALWAYS	The program/script/command *always fails,* regardless of its return code value. **Note:** This keyword is useful for quality assurance testing. It should not be used for production data transfers.

Pre-processing and Post-processing Examples

To illustrate how to successfully use the keywords listed in the previous two tables, study the UNIX operating system and Windows operating system examples shown.

UNIX

This example runs a UNIX shell script (`myScript.ksh`) *before* the data transfer is sent. The script is considered successful if its return code is less than 127.

Figure 7: UNIX Pre-processing Command Syntax Example

```
TRANSFER=(  name=mytransfer
            sendclass=...
            PRE_SEND=[ CMDLINE='sh -x myScript.ksh 2>err.out'
                       RETCODE=127
                       SUCCEEDS_IF_LT
                     ]
```

Windows

This example runs a Windows application to cleanup (delete) data files no longer needed *after* the data transfer is sent. The script is considered successful if its return code equals 0.

Figure 8: Windows Post-processing Command Syntax Example

```
TRANSFER=(  name=mytransfer
            sendclass=...
            POST_SEND=[ CMDLINE='C:\MyPrograms\cleanup.exe /log'
                       RETCODE=0
                       SUCCEEDS_IF_EQ
                      ]
```

Dial-up Connection Keywords (Windows)

These keywords provide a user the capability to access a previously defined Windows Dial-up Networking entries.)

Requirement: You can use these keywords only on Windows 95/98/NT/2000 operating systems.

Table 19: Dial-up Connection Keywords

Keyword	Keyword Values	Description and Usage of Keyword
If your first dialer selection fails to connect, then EasyAccess2000 invokes your second choice. Auto-dialer reports the dialer progress in the main transfer window, and so that it can be invoked during a restart. **Restriction:** If `DIAL=` and `DIAL_PROGRAM=` keywords are both specified, then the Windows Dialup Networking program (`DIAL=`) is used. If `BACKUP_DIAL=` and `BACKUP_DIAL_PROGRAM=` keywords are both specified, then the Windows backup Dialup Networking program (`BACKUP_DIAL=`) is used.		
`DIAL=`	**Name**	Text name of one of the Dial-up Networking entries you have previously set up on your computer.
`DIAL_PROGRAM=`	**X:\Path\ File**	File specification (qualified file name) of a Dialer program that you want started in place of the Windows Dialup Networking application. **Example:** the AT&T Global Network Services Dialer program, `IDIALER.EXE`, (plus directory path) can be specified.
`BACKUP_DIAL_=`	**Name**	Name of a backup Dial-up Networking entry you have previously defined on your computer. The backup Dial entry is used if the primary Dial-up Networking entry or Dialer program fails to connect.
`BACKUP_DIAL_ PROGRAM=`	**X:\Path\ File**	File specification (fully qualified file name) of a Backup Dialer program that you want launched instead of the Windows Backup Dialup Networking application. The backup Dialer program is used if the primary Dial-up Networking or Dialer program fails to connect. **Example:** the AT&T Global Network Services Dialer program, `IDIALER.EXE`, (plus directory path) can be specified.
`AUTODIAL=`	**Y** or **N**	`AUTODIAL=Y`, automatically dial the Dial-up Networking entry with the `DIAL=name` if no dial-up connection is active.
`AUTODISCONNECT=`	**Y** or **N**	`AUTODISCONNECT=Y,` automatically disconnect the current connection when EasyAccess2000 program finishes.
`TIMEOUT=`	**Integer (secs)**	How long EasyAccess2000 should wait for a dial attempt to connect before it decides the attempt has failed.

Auto-Retry Keywords

These keywords control how EasyAccess2000 attempts to complete a data transfer once an initial attempt is unsuccessful.

Table 20: Auto-Retry Keywords

Keyword	Keyword Values	Description and Usage of Keyword
`RETRY=`	**Y** or **N**	Enable (`RETRY=Y`) or disable (`RETRY=N`) Auto-Retry. The default is to have Auto-Retry disabled.
`MAX_RETRY=`	**Integer**	Number of times to attempt a file transfer.
`RETRY_DELAY=`	**Integer**	Number of seconds to wait between data transfer retry attempts.

Trading Partner Address Book - Changing Entry Keywords

These keywords help you change (add or modify) an entry in your Trading Partner Address Book.

Table 21: Trading Partner Address Book – Change Entry Keywords

Keyword	Keyword Values	Description and Usage of Keyword

`TPBOOK=`	**Y** or **N**	`TPBOOK=(…)` specifies an entry in the Trading Partner Address Book is to be added or changed.
		Example: Command-file entry to create/modify a Trading Partner:
		`TPBOOK=( NAME=MyPartner`
		`NETWORK1=”QRS eLink”`
		`MAILBOX1=MyPartnersMailbox`
		`NETWORK2='IGN-I/E SSL'`
		`MAILBOX2=CMAP.MyPartnersIGNAccount )`

These keywords define the Trading Partner entry changed by the `TPBOOK=` keyword.		
`NAME=`	**Text**	Trading Partner name to be added or changed.
`NETWORK1=`	**Text**	Primary Network used when sending or receiving from Trading Partner
`MAILBOX1=`		Primary Mailbox, user ID, or login name for the Trading Partner on the Primary Network.
`NETWORK2=`		**(Optional)** Backup Network used when sending or receiving from this Trading Partner. The Backup Network is used only if a transfer fails using the Primary Network and Mailbox and Auto-Retry is enabled (`RETRY=Y`). The Backup Network is used for the last half of the specified transfer retries if Auto-Retry is enabled (`RETRY=Y`) and `MAX_RETRY=` is two or greater.
`MAILBOX2=`		**(Optional)** Backup Mailbox, user ID, or login name for the Trading Partner on the Backup Network.

Using EasyAccess2000 Applications

Cancel an Active Data Transfer

Once a data transfer starts running, you may want to cancel the transfer. EasyAccess2000 can attempt to cancel the transfer.

Requirement: This is applicable only to data transfers run from the command-line interface.

To cancel an in-progress data transfer

1. Create the file name `cancel.fil` in the EasyAccess2000 `temp` subdirectory folder.
2. The EasyAccess2000 program terminates the transfer, if possible. If it is able to respond, EasyAccess2000 communicates a return code with a value of 2.

Command-line Interface Examples

These examples show you what you would type at the command-line interface to run a utility or the EasyAccess2000 CLI application program.

Create New Data Transfers from the Command-line Interface

(**1**) This example creates a new network definition and a send data transfer.

```
ea2kw95c NETWORK=FEDEXNET
FTPUSERID=CPINC
FTPPASSWD=PASSWORD
NETSTYLE="FEDEXNET"
TRANSFER=(name="A SEND TEST" send=c:\autoexec.bat sendaprf=comptest
senduserid=compressp)
```

(**2**) Use this example to create new receive data transfer.

```
ea2kw95c TRANSFER=(name="big rec1" receive=incoming\autoexec.bat
receiveaprf=comptest receiveuserid=compressp)
```

Example Data Transfers

(**3**) Receive a binary file (in the directory `d:\custout\`) from the server using the file transfer protocol (FTP) over an IBM Global Network-Information Exchange (IGN-IE) and save the transfer information (for the transfer named ‘`my receive`’) in the `easyacc.ini` file.

```
cd \
cd easyacc6
ea2kw95c network=”ign-i/e ssl” ftpuserid=acct.userid ftppasswd=passwd reset
“transfer=(name=’my receive’ receive=d:\custout\ crlf=y)” save
```

(**4**) Send a compressed ASCII text file (file name `d:\custout\eaftplog.txt`) from the server using the file transfer protocol (FTP) over an IGN-IE SSL network and save the transfer information (for the transfer named ‘`my transfer`’) in the `easyacc.ini` file.

```
cd \
cd easyacc6
ea2kw95c network=”ign-i/e ssl” ftpuserid=acct.userid ftppasswd=passwd reset
“transfer=(name=’my transfer’ send=d:\custout\eaftplog.txt
senduserid=acct.userid sendclass=xtro sendascii=y crlf=y compress=y)” save
```

(**5**) Use the stored transfer in example **#3** (‘`my receive`’) with a different user ID and password.

```
ea2kw95c network=”ign-i/e ssl” ftpuserid=cmap.cpinc06 ftppasswd=alan1b reset
“transfer==’my receive’
```

(**6**) Use and revise the stored transfer in example **#4** (‘`my receive`’) with a different user ID and password. Save the new user ID and password in the `easyacc.ini` file.

```
ea2kw95c network=”ign-i/e ssl” ftpuserid=cmap.cpinc06 ftppasswd=alan1b reset
“transfer=(name=’my transfer’ send=d:\custout\eaftplog.txt
senduserid=cmap.cpinc06 sendclass=xtro sendascii=y crlf=y compress=y)” save
```

Send Multiple Files

(**7**) Send a license file using the defined data transfer procedures SEND2 and SEND3.

```
ea2kw95c network=”ign-i/e ssl” ftpuserid=cmap.cpinc04 ftppasswd=qa1test reset
transfer=(name=’SEND2 send=c:\easyacc7\license.txt senduserid=cmap.cpinc04
sendclass=CSM secure=y compress=y) transfer=(name=’SEND3
send=c:\easyacc7\license.txt senduserid=cmap.cpinc04 sendclass=send3)
```

Send EDI Transfer

(**8**) Define and send an EDI data transfer (`sendedi` keyword) with the name `MyEDI` and ‘`SEND TEST`’.

```
ea2kw95c network=”IGN-I/E SSL” ftpuserid=cmap.cpinc04 ftppasswd=qa1test reset
“transfer=(name=’MyEDI sendedi=.FilePath/FileNameToSend)” transfer=(name=’SEND
TEST’sendedi=
/EXT_HD/users/cmercer/EA31BTRD/easyacc/outgoing/out/out/x12002unafil.edi) SAVE
```

Receive EDI Transfer

(**9**) Define and receive an EDI data transfer with the name `RecEDI`. Run a Korn Shell script after the file is received to perform some post processing. *This excellent example shows how to use delimiters to structure your keywords.*

```
ea2kw95c network=”IGN-I/E SSL” ftpuserid=cmap.cpinc04 ftppasswd=qa1test reset
“transfer=(name=’RecEDI receiveedi=.FilePath/FileNameToReceive)”
transfer=(name=’POST RECEIVE TEST’receivededi=./incoming/test.txt
post_receive=[cmdline=’./maint/AtaitPostRec.ksh’ succeeds_always])” SAVE
```

Query Mailbox for Available Files

(**10**) These are two examples of how to query a network mailbox for a list of available files. The list of files is written to the desktop file name `easyacc6\temp\list.fil`.

CAUTION No other transfers are run when the EasyAccess2000 `QUERY_LIST` keyword is specified -- it supersedes the execution of all send and receive transfers.

```
ea2kw95c “network=BTRADEDFW002” ftpuserid=COMPRESSP ftppasswd=PASSWORD reset
QUERY_LIST
```

```
ea2kw95c “network=IGN-I/E SSL” ftpuserid=cmap.cpinc04 ftppasswd=PASSWORD
QUERY_LIST
```

Change Password (FTP Systems Only)

(**11**) These are two examples of how to change a password for an FTP user ID.

```
ea2kw95c “network=BTRADEDFW002” ftpuserid=COMPRESSP
ftppasswd=oldpassword/newpassword/newpassword
```

```
ea2kw95c “network=IGN-I/E SSL” ftpuserid=cmap.cpinc04
ftppasswd=oldpassword/newpassword
```

Importing Encryption Keys

(**12**) Import the encryption keys for an IGN-I/E SSL network that uses the 8-character approval code.

```
parsepfx c:\keys\ign\pfxign\keys\cpinc04.003 21171135
```

(**13**) Import the encryption keys for a non-SSL network that uses the 16-character approval code.

```
>import export.rtm runtime key=0123456789ABCDEF privkey=security
```

Overriding Configured Settings

(**14**) You can override your defined transfers by listing only the ones you wish to run during a single session. For example, you could have four data transfers listed in your **`easyacc.ini`** and **`exfer.ini`** files, but only want to run two transfers. To accomplish this, you would type these two commands:

```
ea2kw95c “transfer=MY SEND TRANSFER” reset
ea2kw95c “transfer=GET ALL INV FILES” reset
```

EasyAccess2000 runs the two specified transfers.

Examples of Stored Transfers

From the command line within the subdirectory where EasyAccess2000 is installed, type **ea2k*c** and press [**Enter**]. The asterisk character “*” is the variable for the correct version of EasyAccess2000 command-line interface. See *Table 6, EasyAccess2000 Operating Systems Applications*, for the name of the command-line interface appropriate for your operating system. For example, if you are running Windows 95, you would type “**ea2kw95c**” to run the command-line interface for Windows 95/98.

EasyAccess2000 command-line interface runs the stored transfers located in the **`easyacc.ini`** and **`exfer.ini`** files that you specified using the EasyAccess2000 GUI or a text editor.

Use Existing Send Transfer

```
ea2kw95c TRANSFER=SENDINVOICES reset
```

Or

```
ea2kw95c “transfer=send invoices” reset
```

Use Existing Receive Transfer

```
ea2kw95c “TRANSFER=RECEIVE TEST” reset
```

Use Existing Combined Send and Receive Transfer

```
ea2kw95c “TRANSFER=PUT AND GET ALL FILES” reset
```

Command File Example

This example of an EasyAccess2000 command file defines a number of features that the program’s keywords implement. The syntax (rules) of using these keywords is also illustrated. As always, the pound sign (#) acts as a comment marker within the file and blank lines are ignored. This helps you document your work so that other bTrade.com employees can use this file also.

Figure 9: Example EasyAccess2000 Command File

```
# (Remove leading '#' character to activate any given line)

NETWORK=Btrade.com       # Select the network to use (do only once)
USERID=myUserId          # Override the server login userid
```

```
PASSWD=myPassWord        # Override the server password


# Create some transfers, invoke other transfers already stored
# Note the use of quotes surrounding the data transfer name:

TRANSFER="MY RECEIVE TEST"  # Run 'MY RECEIVE TEST' transfer



# You can use single or double quotes, square brackets, curly braces, or
# parentheses for transfer creation, for pre/post processing
# specification, and for TPBOOK= changes also.
# (Trading Partner Address Book changes)

transfer=(                   # Create a new transfer and then execute it
    name="MY SEND TEST"      # Name is REQUIRED!
    send=c:\autoexec.bat
    senduserid=CPINC03       # Server is case sensitive: userid in caps!
    sendclass=DOMINV         # Receive all files in DOMINV class
    COMPRESS=Y SENDASCII=Y CRLF=Y FILTER=Y SECURE=N
    pre_send= [cmdline='dir *.*' retcode=0 succeeds_always]
    post_send= [cmdline='sh -x /home/user/cleanup.ksh -h -l=60'
                retcode=0
                succeeds_if_eq]
               )             # End transfer creation

Transfer= (Name=SendInvoice send=c:\inv\invoices.txt SendUserId=CPINC03
           SendAPRF=INV)

transfer= (name=ReceiveINV receive=c:\inv\new_inv.txt receiveuserid=CPINC03
receiveclass=INV autoext=y ascii=y append=n)



# Obtain a list of available files from the server.
# The file list is written to file myaudit.log. If the auditFile keyword
# is not specified,  list is written to the default file, list.fil
# in the EasyAccess 'temp' directory.

# No other transfers are executed during the program run if
# the "queryList" keyword is specified -- it supersedes the execution
# of all send and receive transfers.

queryList queryFile=myFiles.lst


# Get an audit report from the server showing all files sent and received
# from and to a client (current login) during 11/1/1999 through 12/1/1999

# Audit report is written to audit.log in the EasyAccess "temp" directory.

# No other transfers will be executed during the program run if
# the "receiveAuditLogs" keyword is specified -- it supersedes the execution
# of all send and receive transfers.

receiveAuditLogs
    auditStartDate=19991101
    auditEndDate=19991201
    auditFile=myaudit.log
```

```
# Create a public/private key pair, generate a certificate request, and
# send the request to the configured Certification Authority for approval
# (as specified in the SECURITY section of the easyacc.ini file).

# Additionally, specify a pass-phrase location to store the key-encrypting
# key, used to provide security-token capabilities.
# No other transfers are executed during the program run if
# the "genkeys" keyword is specified -- it supersedes the execution
# of all send and receive transfers.

genkeys passloc=[a:/mytoken.txt]



# Create and execute a transfer to receive the user's Security Runtime Files
# (previously generated by the configured Certification Authority
# as specified in the SECURITY section of the easyacc.ini file). Once the
# Security Runtime Files are received, they are installed.automatically

# No other transfers will be executed during the program run if
# the "receive_runtimes" keyword is specified -- it supersedes the execution
# of all send/receive transfers.

receive_runtimes



# Create some Trading Partner Address Book entries, and use one in a transfer
# The network2 and mailbox2 TPBOOK keywords are optional.

TPBOOK=( name=myPartner network1=Btrade.com mailbox1=MyPartnersMailbox )

# Note you can use square brackets, or parentheses for transfer creation,
# and for pre/post processing specification, and for TPBOOK= usage too.

TPBOOK=[ name=myOtherPartner
         network1=Btrade.com
         mailbox1=MyOtherPartnersMailbox
         network2="IGN-I/E SSL"
         mailbox2=CMAP.MyOtherPartnersIgnMailbox
       ]



# This transfer tries to send to myOtherPartner on the Btrade.com network.
# If this fails, it switches to the IGN-I/E SSL network (since auto-retry
# is enabled below.)

Transfer= (Name=ShowOff send=c:\inv\invoices.txt SendUserId=myOtherPartner
           SendAPRF=ShowOff)



# Illustrate use of Dial-Up and Backup Dialup Networking Keywords.
# (Windows operating systems only)

DIAL="Dial Postal"    # Name an existing Windows Dial-Up Networking entry
# If the "Dial Postal" DialUp Networking entry fails to connect, then
# start the following Dialer program to try an alternative connection:

BACKUP_DIAL_PROGRAM='C:\Program Files\AT_T_GlobalDialer\IDIALER.EXE'
AUTODIAL=Y                # Dial before trying to connect!
AUTODISCONNECT=Y          # Hang up when program is finished
TIMEOUT=180               # If no connection in 3 minutes, then failed
```

```
# Illustrate the usage of the Auto-Retry Keywords.

RETRY=Y                    # Enable auto-retry
MAX_RETRY=2                # Retry twice after initial failure
RETRY_DELAY=10             # Delay 10 seconds between retries
```

Using IEBASE Option

IBM Expedite Base (IEBASE) /AIX is a communications component of IBM Interchange Services for e-business that runs in the AIX Version 4.2.1 environment. Expedite Base/AIX is used to exchange electronic data with trading partners via Information Exchange, the mailbox component of IBM Interchange Services. IEBASE uses Comm-Press2000 as its underlying compression and encryption software.

IEBASE functionality is supported, although direct operations using batch file commands or the command-line interface are easier to implement. To use the **`IEBASE`** option, type **ea2k*c iebase** and press [**Enter**].

`*` - denotes the actual characters needed for the EasyAccess2000 command-line interface application that applies to your operating system.

IEBASE reads the commands listed in the `BASEIN.MSG` file. See the next major section of the *EasyAccess2000 Configuration Guide* for specific instructions on building this file.

Using the IEBASE.EXE Application

Overview

EasyAccess2000 can be used to interpret `BASEIN.MSG` command files and perform IBM Expedite Base batch transmissions with the *FedEx Net* (FEDEXNET) and *IBM Global Network-Information Exchange* (IGN-IE) networks. Using the `IEBASE.EXE` program provided with EasyAccess2000, you can use EasyAccess2000 as a drop-in replacement for the IBM Expedite Base application. (See restrictions described below.)

The `IEBASE.EXE` application sends and receives from multiple mailboxes during one session, plus performs multiple computer logons during a single session. The `IEBASE.EXE` program acts as a batch front-end to the EasyAccess2000 application.

The `IEBASE.EXE` application reads the `EASYACC.INI` file to determine the current network located in the IDENTIFY section. The `IEBASE.EXE` application defaults to the FTP interface for IGN-IE. The IGN-IE account, userid, and password are read from `BASEIN.PRO`. FEDEXNET user ID and password must be configured in the `FEDEXNET` section of the `EASYACC.INI` file.

To summarize all this information:

Table 22: IEBASE.EXE Application

Network	EASYACC.INI Section	Determine this Information
IGN-I/E	`IDENTIFY`	Current network style for data transfer
IGN-I/E	`n/a`	`BASEIN.PRO` file – Account, userid, and password
FEDEXNET	`FEDEXNET`	`USERID` and `IEPASSWORD`

Commands

The `IEBASE.EXE` application reads the `BASEIN.MSG` file and converts the IBM Expedite Base-style commands to EasyAccess2000 Stored Transfers. Current **Expedite** commands that are recognized include `START, SEND, SENDEDI, RECEIVE,` and `RECEIVEEDI`.

Table 23: IEBASE Commands

Recognized IEBASE Commands	Command Description
`Start`	Logon to the network as a new user. Ignore account field for FEDEXNET networks
`SEND` `SENDEDI`	Specify the files to send.
`RECEIVE` `RECEIVEEDI`	Specify the files to receive.

Examples

START Command

```
start ACCOUNT(fecp)  userid(123456789) IEpassword(PASSWORD);
```

If you are using FEDEXNET, IEBASE.EXE reads only the USERID and IEPASSWORD to logon.

SEND and RECEIVE Commands

```
send fileid(c:\FEDEX.EDI) userid(GOFEDEXASYNC) class(revp570);

receive fileid(c:\DOMinv.FIL) class(DOMinv);
```

Multiple Mailboxes

EasyAccess2000 can send and receive from multiple mailboxes during a single session by providing sequences of START, SEND, and RECEIVE commands in the BASEIN.MSG file.

```
#
start account(FECP) userid(123456789) iepassword(password);
send fileid(c:\fedex.dat) class(revp570) userid(gofedexasync);
send fileid(c:\remit.dat) class(remit) userid(gofedexasync);
receive fileid(c:\easyacc6\incoming\dominv.FIL) class(dominv);
receive fileid(c:\easyacc6\incoming\intlinv.fil) class(intlinv);
#
#
start account(FECP) userid(CA531345678) IEpassword(password);
send fileid(c:\fdnx1010.fil) class(revp570) serid(gofedexasync);
#
#
start account(FECP) userid(PR098765432) IEpassword(password);
send fileid(c:\fedex.dat) class(revp350) userid(gofedexasync);
receive fileid(c:\easyacc6\incoming\prtrk.FIL) class(bulktrk);
#
```

RECEIVE FILEID(C:\easyacc6\incoming\INTLINV.FIL) CLASS(INTLINV); ← FEDEXNET Example

Notice the different user IDs and mailboxes in each START command.

CAUTION

If you are communicating with FEDEXNET, it is important to note that the FTP server is case-sensitive and lower-case characters must not be used when typing the parameters for the BASEIN.MSG file. *The syntax for this file must be followed exactly as indicated in the example.*

Scheduling Automated Data Transfers

Creating A Batch File to Run Unattended

Many users must access multiple mailboxes and run data transfers without operator intervention. EasyAccess2000 supports a batch execution option. This section explains how to perform data transfers in the multiple-mailbox batch file. (See following example.)

Example Batch File for Multiple Mailbox Access

EasyAccess2000 can be run by using:

- DOS batch programs
- Command-line programs within other programs
- Calling IEBASE.

An example DOS batch program (simplest) would use these keywords to run a stored data transfer:

```
cd\easyacc6
ea2kw95c “transfer=SEND REMITTANCE DATA” reset
```

To run EasyAccess2000 unattended using IEBASE:

1. Start a text editor application.
2. Create a new BATCH file. For example, `EABATCH.BAT`.
3. Insert your entries. You would create a line in your batch file for each mailbox. See “`TP1.MSG`” on the next page for a detailed explanation of the file.

```
Copy TP1.msg  Basein.msg
```

Batch File Components

```
Copy baseout.msg TP1log.msg
Copy TP2.msg  Basein.msg
IEBASE.EXE
Copy baseout.msg TP2log.msg
Copy TP3.msg  Basein.msg
IEBASE.EXE
Copy baseout.msg TP3log.msg
Rem  End of File
```

File Definitions

Table 24: Batch File Components

File Names / Other	Description of File
`COPY`	Basic DOS copy command
`TP1.MSG, TP2.MSG, TP3.MSG`	Each mailbox to be automated requires an individual Trading Partner `msg` (message) file (`TP1.msg`). This file contains all the relevant information pertaining to that mailbox (user) and which transfers take place for that mailbox. You would create a line in your batch file for each mailbox. See “Mailbox.MSG Message Files” below for a detailed explanation of this file.
`BASEIN.MSG`	This text file contains the appropriate transfers and identifies each mailbox to the network.
`IEBASE.EXE`	Application program that processes the transfers.
`BASEOUT.MSG`	The `IEBASE` program generates this text file after each session and reports the status of the run transfers. It is replaced after each application run.

`TP1LOG.MSG,`
`TP2LOG.MSG,`
`TP3LOG.MSG`
	By copying the `BASEOUT.MSG` file to each mailbox log file, a review of this file can confirm the transfer status for that mailbox. This is especially helpful in the case of communication and hardware failures.

Mailbox.MSG Message Files - TP1.MSG

```
START ACCOUNT(FECP) USERID(TP1MAILBOX) IEPASSWORD(PASSWORD);
send  fileid(C:\PATH\FNAME) class(REMIT) USERID(GOFEDEXASYNC);
```

Table 25: Description of Mailbox Message Files

Item	Description
`START`	Identifies the user to the network—user ID, password, and accounting information.
`ACCOUNT(FECP)`	Accounting information his is not used by FEDEXNET, but it must be typed anyway
`USERID(`*`TP1MAILBOX`*`)`	Replace *`TP1MAILBOX`* with a valid mailbox number.
`IEPASSWORD(`*`PASSWORD`*`)`	*`PASSWORD`* should be replaced with the valid password for this mailbox number.
`SEND or RECEIVE`	Denotes the type of transfer
`FILEID(`*`DRIVE`*`:\`*`PATH`*`\`*`FNAME`*`)`	*`Drive`*—drive letter designator *Path*—directory file path *Fname*—the filename being sent or received
`CLASS(REMIT) - APRF`	Valid APRF for the data type you are accessing. Some examples are—`DOMINV`, `INV`, `REMIT`, `REVP570`.
`USERID(`*`GOFEDEXASYNC`*`)`	This value is only needed when you are sending and is the destination or recipient mailbox. For `FEDEXNET`, the value is always `GOFEDEXASYNC`.

easyacc.ini File Reference

This section describes the `easyacc.ini` file and the keyword default values used. The network style you select has the defaults to be used.

CAUTION Changing the `easyacc.ini` file defaults should only be done with the assistance of a bTrade.com Product Support person.

General Comments about easyacc.ini File

- The value "`NONE`" is a valid default string used in several places.
  A blank or null default is indicated by "`-`".

- In the text describing each entry, the syntax $name means the contents of the field name in the `easyacc.ini` file; for example, `$BASENAME` is the contents of the `BASENAME` field in the `easyacc.ini` file.

- An entry of `N/A` indicates the field is not applicable to this EasyAccess2000 software version.

- Many sections are prefaced by network name, allowing each network to contain its own version of a particular section. Prefixing the network name to the section name does this.

  `Example: the RECEIVECLASS` section appears for each network, giving sections in the `easyacc.ini` file like `[FEDEXNET- RECEIVECLASS], [IGN I/E SSL-RECEIVECLASS]`, and so forth.

  This variable is noted in the table below as `[<network-name>-RECEIVECLASS], or <nn>-RECEIVECLASS.`

- Some sections allow an open-ended list of entries. These are denoted by the entry <…list>.

Field Names and Descriptions

Major Keyword Sections of the File

Table 26: EasyAccess2000 easyacc.ini File Contents

Section Name Field Name	Field Description	Format Contents

`[EAPATH]`
`BASEPATH=`
	Stores the current parent directory folder of the EasyAccess2000 application.
	Default: `C:\EasyAcc6\` This is the directory folder in which the `easyacc.exe` and `easyacc.ini` files reside.
		X:\Pathname\

Section Name Field Name	Field Description	Format Contents
[REGISTRATION] **COMPANY=**	User's company name on registration.	Text
NAME=	User's name of record.	Text
[IDENTIFY] **AUDIT_END_DATE=**	Specifies the end date when filtering the audit.log file.	yyyymmdd
AUDIT_START_DATE=	Specifies the start date when filtering the audit.log file.	yyyymmdd
AUTO_RETRY=	Specifies whether automatic dial-up retry is to be used.	Y or N
[IDENTIFY] **MAX_RETRY**	Maximum number of retries to be attempted.	Integer
RETRY_DELAY	Number of seconds to wait between automatic retries.	Seconds
DISABLE_DIALER	DISABLE_DIALER=Y, EasyAccess2000 Client (Windows 95/NT only) completely turns-off theWindows RAS Dialer functionality.	Y or N
NETWORK	Internal field which specifies the network name of the active (current) network.	Text
MULTITHREADED	Specifies whether the product should run as a multi-threaded application on this computing operating system. Specify N if required for your platform.	Y or N
MULTIFILE	Internal flag that specifies whether the program is to process: o Ad-Hoc transfer (MULTIFILE=N) o Set of Stored Transfers in the exfer.ini file (MULTIFILE=Y) o Set of Stored Transfers in the bexfer.ini file (MULTIFILE=B)	Y, N, or B
STARTTIME	Specifies the time when a scheduled data transfer is to begin.	hhmmss
STARTDATE	Specifies the time when a scheduled data transfer is to begin.	yyyymmdd
Logging Levels used for these six keywords: **LOG_MEM=** **LOG_INI=** **LOG_XFER=** **LOG_FTP=** **LOG_EASYACC=** **LOG_THREAD=**	Values are N (no logging), Y (level 3 logging), or an integer value between 1 and 6, which specifies the logging level. Level 6 means very-detailed logging that should be used with care because it introduces substantial processing overhead. If the integer value is prefixed with a '-' (minus sign), the log file is closed after each write. This option preserves log entries in the case of a program *abnormally ends* (ABENDS) on an error. It introduces a great deal of overhead and may impact the speed of program execution. **Caution:** Do not use the '–' (minus sign) option or level 6 for normal operation. Use these for showing logging results to bTrade.com Product support personnel.	Y or N 1 2 3 4 5 6 -1 -2 -3 - 4 -5 -6
LOG_MEM=	Log memory usage. Default is LOG_MEM=N, no logging.	See Above

Section Name Field Name	Field Description	Format Contents
`LOG_INI=`	Log reads and writes to the `easyacc.ini`, `exfer.ini` and `bexfer.ini` files. Default is `LOG_INI=N`, no logging.	`See Above`
`LOG_XFER=`	Log all internal FTP, compression, and decompression program activities. Default is `LOG_XFER=N`, no logging.	`See Above`
`LOG_FTP=`	Log all internal and external FTP activities. Default is `LOG_FTP=6`, no logging.	`See Above`
`LOG_EASYACC=`	Write a general log of the session. Default is `LOG_EASYACC=N`, no logging.	`See Above`
`LOG_THREAD=`	Enable logging in processing threads. Default is `LOG_XFER=N`, no logging.	`See Above`

`[NETWORKS]`
`<list>`
	<…list>
	The list of entries available from which a user can select.
	Format of the list: `number = Network Name`
	
	Example:
	`98=EDIONTHENET`
	`1=IGN-I/E`
	`2=IGN-I/E SSL`
		number = name

`NETWORK=`	Internal field which specifies the network name of the active security network.	Text

`[SECURITY]`
`ADDRESS1=`
`ADDRESS2=`
`COMMONNAME=`
`COUNTRY=`
`LOCALITY=`
`ORGANIZATION=`
`ORGUNIT=`
`PARTICIPANT=`
`POSTALCODE=`
`STATE=`
`TITLE=`
	Keywords Usage:
	o Used to construct the static encryption key
	o Displays as part of Participant Information when using the **Security->Registration** command in the EasyAccess2000 GUI (Restricted to X12.58-enabled networks)
	o Builds a distinguished name in a certificate request using the **Security->Registration**->**GenKeys** command (Restricted to X12.58-enabled networks)
		Text

`APPROVALCODE=`	**Keyword Usage:** o Logon password for first-time logon (Only applies if `LOGONREQUIRED=Y`) Can be superseded by the user after the first logon by entering a new logon password. o Value passed as an argument into the **Import** application during the **Security->Install Certificate** command. (Restricted to X12.58-enabled networks) o Value passed as an argument into the **Compress** application during **Security->Receive Certificates** command. (Restricted to X12.58-enabled networks)	Text

Section Name		Format
Field Name	Field Description	Contents
AUTOUPDATERUNTIME=	AUTOUPDATERUNTIME=Y, automatic call to **Security->Receive Certificates** command during program startup to update the Security Runtime Files. (Restricted to non-IGN SSL and X12.58-enabled networks.)	Y or N
CERTDEST=	Identity of security server directory that receives certificate requests.	X:\Pathname\
EDINAME=	**Keyword Usage:** o (**Requirement:** SSL-enabled networks) Used during SSL negotiation during FTP connect for the following units of work: **Audit**, **Edit Network/Change Network Password**, **Send** or **Receive** files in any file transfer, and **Query Mailbox**. o (**Requirement:** X12.58-enabled networks) Used to construct unique file names used in the generation and receipt of Security Runtime Files. o (**Requirement:** X12.58-enabled networks) Used to construct the header.def file used in the "SECFILE=<pwd>/runtime/header.def" argument to compress application for non-EDI files. May be disabled by setting SECURE=N in the SEND PARMS section. The header.def file contains alias information about the sender and receiver. o (**Requirement:** X12.58-enabled networks) Displayed on the **Participant Information** window using the **Security→Registration** command.	Text
EXPDATE=	Used to determine when the user's trial period has expired. Specifies a number of days after a specific date.	Text
LOGONREQUIRED=	LOGONREQUIRED=Y**,** the EasyAccess2000 GUI **Logon** window is displayed, requiring the user to give a user ID and password.	Y or N
LOGON_PWD=	**Keyword Usage:** **Requirement:** Option LOGONREQUIRED=Y. o If no value is present in the *.ini file, then the **Change Password** check-box is selected and disabled, forcing the user to enter a new password. o Logon password. If no value is present, then the **APPROVALCODE** field is used as the logon password and usage above is applied.	Text
MODULUS=	(**Requirement:** X12.58-enabled networks) Used in the generation of a certificate request. Default MODULUS=512.	Integer
PASSWORD=	Password used to logon to the security server.	Text

Section Name		Format
Field Name	Field Description	Contents
RTMGENERATE=	`RTMGENERATE=Y`, security server supports the automatic generation of Security Runtime Files.	`Y` or `N`
RTMCLASS=	**Keyword Usage:** **Requirement:** Option `RTMGENERATE=Y`. o For security servers that support the automatic generation of Security Runtime Files, this keyword defines the directory folder it is to place the generated Security Runtime Files. o When receiving Security Runtime Files (using the **Security->Receive Certificates** command or `AUTOUPDATERUNTIME=Y`), defines the server directory in which the Security Runtime Files are located.	X:\Pathname\
TELEPHONE=	**Keyword Usage:** o Constructs the static encryption key o Displays in the Participant Information for the **Security->Registration** command (Available in X12.58-enabled networks)	Text
[MAINT] **NETWORK=**	Network used to conduct all maintenance-related transfers (such as, receiving maintenance updates to the software).	Text

The following keywords apply to *all defined networks*.
Each network defined has a main Section and five sub-sections defined by a name.

Example: Keyword subsection structure for each network defined in the `easyacc.ini` file.

```
[NETWORKNAME1]
MainSection_Keyword1=aaaaaa
MainSection_Keyword2=bbbbbb
MainSection_Keyword3= etc.

[NETWORKNAME-DEFAULT_SENDPARMS]
SendParms_Keyword1=dddddd
SendParms_Keyword2=eeeeee
SendParms_Keyword3= etc.

[NETWORKNAME-DEFAULT_RECEIVEPARMS]
ReceiveParms_Keyword1=ffff
ReceiveParms_Keyword2=ggggg
ReceiveParms_Keyword3= etc.

[NETWORKNAME-SENDCLASS]
SendClass_Keyword1=hhhhhh

[NETWORKNAME-RECEIVECLASS]
ReceiveClass_Keyword1=iiiii

[NETWORKNAME-MULTITRANS]
MultiTrans_Keyword1=jjjjjj
```

Section Name Field Name	Field Description	Format Contents
[NETWORKNAME] **AUTO_DIAL=**	AUTO_DIAL=Y, auto-establish a dial-up connection prior to executing any transfer to the FTP server.	Y or N
AUTO_DISCONNECT=	AUTO_DISCONNECT=Y, auto-terminate a dial-up connection when the EasyAccess2000 application ends.	Y or N
CASE=	All communication with the server is to be converted to upper case. If not, then normal case sensitivity is assumed CASE=U – convert and send in upper-case format. CASE=L – send data to the server unchanged.	U or L
CONTROL_PORT=	Port number that EasyAccess2000 uses to communicate with the FTP server.	Integer
DIAL_ENTRY=	Which Windows dial-up network connection to use when communicating with the server.	Text
HOSTIPNAME=	Primary IP address or domain name for the server.	Host Name IP Address
HOSTIPNAME2=	Backup IP address or domain name for the server.	Host Name IP Address
MAX_AUTO_DIAL_DELAY =	Timeout value for attempting to establish a dial-up connection. Default is MAX_AUTO_DIAL_DELAY=180 If the connection has not been established after the time out period, any pending transfers are not run and an error is reported.	Integer (Seconds)
NAME=	Network name that for which this and the next four subsections of the easyacc.ini file describe.	Text

Section Name		Format
Field Name	Field Description	Contents
[NETWORKNAME] **NETWORKSTYLE=**	Network communications style used by the server: **FTP Communication Networks:** o GENERIC-FTP o GENERIC-DOS o GENERIC-SSL o ICC-NET (Internet Commerce Corporation's ICC.NET service) o IGN-IE o FEDEXNET o WALMART o EAFTP o GEIS MARK III o EDI*Express (GEIS) o EDISWITCH (GEIS) o CONNECTMAIL o Sterling-Commerce (Sterling) o MCI-EDI*NET o QRS-ELINK **SMTP/POP3 Mail Servers:** o EDI-INT o GISB-CLIENT o GISB-SERVER **Compression ,encryption, and archieving (no file transfer)** o LOCAL-ARCHIVE **Undocumented** o CONNECTMAILBOX o DATAGUARD o EDIONTHENET o FEDEXNET X12.58 o GENERIC-MVS o IGN EMEA SSL o IGN EMEA SSL o IGN EMEA X12.58 o IGN-I/E SSL o IGN-I/E WITH COMPRESSION o IGN-I/E o IGN-I/E X12.58	Text
PASSIVE=	PASSIVE=Y, FTP session is to use passive mode.	Y or N
PASSWORD=	Server logon password.	Text
SSL=	SSL=Y, SSL 3.0 session-level security is to be used during the FTP session.	Y or N
SECURITYMENU=	Security Menu on the main EasyAccess2000 GUI window is enabled: o SECURITYMENU=N, no menu. o SECURITYMENU=Y, full security menu. o SECURITYMENU=I, replace menu with the ParsePFX utility option to import Security Runtime Files for IGN-I/E-SSL networks.	Y, I, or N

Section Name Field Name	Field Description	Format Contents
SITEDELAY=	Duration between FTP commands to the FTP server when processing runtime transactions (certificate requests and receiving Security Runtime Files). Required for some FTP servers to avoid communication problems.	Integer (Seconds)
SUNIQUE=	EasyAccess2000 uses the **Store Unique** command (RFC-959) for putting files onto an FTP destination server. This command instructs the server to assign a random, unique name to the file as it creates and writes the file. Although RFC-959 specifies the formats of this command, two forms are commonly used: ○ Format 1 = STOU <filename> (takes filename as an argument) ○ Format 2 = STOU (no argument – RFC-compliant format) **SUNIQUE=0,** does not generate a unique file name (puts file on server using source file name). **Caution**: If you have already put a file with the same name, the FTP put command may fail and EasyAccess2000 alerts the user that a file by that name already exists. **Solution**: Change the filename that you are sending before executing the transfer **SUNIQUE=1,** issues the STOU command using a unique file name as an argument (for example, `STOU <filename>`). **SUNIQUE=2,** issues STOU command without an argument (for example, `STOU`). For FedEx users, EasyAccess2000 defaults to `SUNIQUE=2` in the appropriate network section. For some networks all data and commands may be encrypted and some proxy servers may not function as designed under this configuration.	0, 1, 2
UPDATERUNTIME=	`UPDATERUNTIME=Y`, user is receiving Security Runtime Files using the **Security->Receive Certificates** command.	Y or N
USERID=	Server logon user ID.	Text

Network Subsections

Table 27: Network Subsections in the Easyacc.ini File

Network Name-Section Name Keyword	Field Description	Format Contents
These entries define the default behavior for the **Send** portion of a transfer and are network specific. **nn** denotes the network name that is the prefix for the subsection name.		
[nn-DEFAULT_SENDPARMS] **ASCII=**	`ASCII=Y`, translates the data to `ASCII` or `EBCDIC` (if necessary) depending on the computer operating system where the data is decompressed.	Y or N

Network Name-Section Name Keyword	Field Description	Format Contents
[nn-DEFAULT_SENDPARMS] **COMPRESS=**	COMPRESS=Y, compress file before transmission to server	Y or N
[nn-DEFAULT_SENDPARMS] **CRLF=**	CRLF=Y, convert delimiter characters (for example, line feeds or carriage return/line feed pairs) into record separators.	Y or N
[nn-DEFAULT_SENDPARMS] **FILTER=**	FILTER=Y, invokes the filter algorithm described in *request for comment* (RFC-1113) to convert the compressed data from binary into text format. Filtered data is always transmitted as a text file.	Y or N
[nn-DEFAULT_SENDPARMS] **SECURE=**	SECURE=Y, encrypt a file before it is sent to the server.	Y or N
These entries define other Compression Options that can be added by manually editing the Easyacc.ini file and inserting keywords (format KEYWORD=Y) to make the compression option active.		
[nn-DEFAULT_SENDPARMS] **DELETE_AFTER_SEND=**	DELETE_AFTER_SEND=Y, delete the file after receiving a acknowledgment of successful transmission.	Y or N
[nn-DEFAULT_SENDPARMS] **DELIMIT=**	DELIMIT=n, insert an appropriate delimiter character to obtain records with n characters each.	Integer
Defines a list of classes or *Application Reference Fields* (APRFs) to which the user can send. For some networks, the user can send *only* to these classes, for other networks, the user can use these classes or create their own list.		

[nn-SENDCLASS]
name<nn>=
	Example list of network APRFs (classes):
	[MYEXNET-SENDCLASS]
	CLASS1=BLKTR
	CLASS2=CRBLKTR
	CLASS3=EDRRQAP
	CLASS4=EFTCRCV
		Text

These entries define the default behavior for the **Receive** portion of a transfer and are network specific. **nn** denotes the network name that is the prefix for the subsection name.		
[nn-DEFAULT_RECEIVEPARMS] **APPEND=**	APPEND=Y, received file is to be appended to the client's target file.	Y or N
[nn-DEFAULT_RECEIVEPARMS] **AUTOEXT=**	AUTOEXT=Y, Specifies that the received file is to be given a unique name by auto-extending the client's target file name. (Note: User does not have control of file extensions appended to the file name.	Y or N
[nn-DEFAULT_RECEIVEPARMS] **ASCII=**	ASCII=Y, decompression ASCII option is to be used.	Y or N
[nn-DEFAULT_RECEIVEPARMS] <…**name>=**	name=Y, defines other Compression Options that can be added by manually editing the Easyacc.ini file and inserting keywords (format name=Y) to make the decompression option active.	

Network Name-Section Name		Format
Keyword	Field Description	Contents
Defines a list of classes or *Application Reference Fields* (APRFs) from which the user can receive. For some networks, the user can receive *only* to these classes, for other networks, the user can use these classes or create their own list.		

[nn-RECEIVECLASS]
name<nn>=
	Example list of network APRFs (classes):
	[MYEXNET-RECEIVECLASS]
	CLASS1=ADDR
	CLASS2=BULKTRK
	CLASS3=DISPCONF
	CLASS4=DISPRPT
		Text

[nn-MULTITRANS] **name=**	List of stored transfers to run when the IDENTIFY section keyword MULTIFILE=Y or MULTIFILE=B. o MULTIFILE=Y, set of Stored Transfers in the exfer.ini file o MULTIFILE=B, set of Stored Transfers in the bexfer.ini file	Names

Glossary

All glossary terms in blue are recent additions to the bTrade.com terminology glossary.

A-B

AES

Advanced **E**ncryption **S**tandard. A new Federal Information Processing Standard (FIPS) that specifies an encryption algorithm(s) capable of protecting sensitive government information well into the twenty-first century. The U.S. Government will use this algorithm and the private sector will use it on a voluntary basis.

algorithm (cryptographic)

A clearly specified mathematical computation process; a set of rules that gives a prescribed result.

alias

A name that is more easily remembered for a network or software object. Example: Your PC client name or a server directory folder.

APRF

Application **R**eference **F**ield (Class). A set of classes that an EDI application can receive data from or to which it can send EDI data. EasyAccess2000 uses these classes to filter EDI data during stored transfers.

AS3 [Internal bTrade]

Applicability **S**tatement **3** describes how to use Secure FTP protocol with Control Ports to perform secure data transfers with command over data ports.

asymmetric encryption

An algorithm that uses two mathematically related, yet different key values to encrypt and decrypt data. One value is designated as the private key and is kept secret by the owner. The other value is designated as the public key and is shared with the owner's trading partners. The two keys are related such that when one key is used to encrypt data, the other key must be used for decryption. See *public key*, *private key,* and *trading partner*.

Batch Mode

EasyAccess2000 operation where a list of transfers is executed as a single EDI transmission and reception.

bTrade.com

bTrade.com uses the Internet to connect the business applications of complex e-Business trading communities, implementing solutions at speeds unprecedented in the market.

C-D

certificate

A public key certificate. Certificates are issued by a certification authority (CA), which includes adding the CA's distinguished name, a serial number and starting and ending validity dates to the original request. The CA then adds its digital signature to complete the certificate. See *CA* and *digital signature*.

Certificate File

A SecureManager2000 runtime file containing the public keys of all the trading partners who wish to exchange secure data. The public keys are stored in a Certificate format that is defined according to the ANSI X.509 standard. Certificates contain the (**1**) unique public key owner's Distinguished Name, (**2**) a copy public key copy, and the (**3**) starting and ending validity dates.

certificate request

An uncertified public key created by a trading partner as part of the Rivest Shamir Adleman (RSA) key-pair generation. The certificate request must be approved by a certification authority (CA), which issues as a certificate, before it can be used to secure data. See *CA*, *public key*, *RSA*, *trading partner*, and *uncertified public key*.

ciphertext

Another name for encrypted data.

Comm-Press2000™

bTrade.com's underlying core utilities that allow you to compress, encrypt, authenticate, and assure data files for cross-platform file transfers over public and private networks.

CRLF resolution

A run-time option in Comm-Press2000™ which compensates for the differences in how records are stored on platforms that use highly structured, record-oriented I/O (for example, MVS, OS/400, and VMS) and platforms that use a continuous stream of bytes.

Dataguard™

EDI software product from Sterling Commerce.

delimiter

A field separator (for example, comma, tab, or other defined character) within a data record.

decryption

The process of transforming ciphertext into plaintext.

distinguished name

A set of data that identifies a real-world entity, such as a person in a computer-based context.

DLL

Dynamic **L**ink **L**ibrary. A collection of small programs, any of which can be called when needed by a larger application that is running in the computer

E

Easyacc.ini

The complex initialization file EasyAccess2000 uses to configure stored data transfers between trading partners (possibly on different operating systems).

EasyAccess2000™

EasyAccess2000 is a secure data communications bTrade.com product that links customer business applications and processes to different IP gateways, portals, and servers used by e-Business trading communities. EasyAccess2000 software (**1**) displays critical audit information on a real-time basis, (**2**) is distributed from bTrade.com's Internet, (**3**) employs high-performance data transmissions, and (**4**) uses state-of-the-art data compression to secure session transactions via the Internet.

EBCDIC

Extended **B**inary-**C**oded **D**ecimal **I**nterchange **C**ode; An IBM code for representing characters as numbers. Although widely used on large IBM computers, most other computers, including PCs and UNIX workstations, use ASCII codes.

EDI

Electronic **D**ata **I**nterchange: The inter-organizational, computer-to-computer exchange of business documentation in a standard, machine-processed format; using national or international standards. See also ANSI X12 and EDIFACT.

EDIFACT

United **N**ations **E**lectronic **D**ata **I**nterchange **f**or **A**dministration, **C**ommerce, and **T**ransport. International standard set by the UN and administered in the U.S. by DISA. This standard has been widely implemented in western Europe.

EDI-INT

Electronic **D**ata **I**nterchange-**I**nternet **I**ntegration. An active working group of the Internet Engineering Task Force that focuses on method for packaging the EDI X12 and UN/EDIFACT transactions sets in a MIME envelope. This group goes beyond RFC-1667 and addresses additional requirements for obtaining multi-vendor, inter-operable service, over and above how the EDI transactions are packaged, These currently revolve around security issues such as EDI transaction integrity, privacy, and non-repudiation.

EDI name

A unique identifier used by the Comm-Press2000 software and public networks for addressing and routing EDI files.

encryption

The process of transforming plaintext into an unintelligible form (ciphertext) such that the original data either cannot be recovered (one-way encryption) or cannot be recovered without using an inverse decrypting process (two-way encryption).

expEDIte/PC®

An IBM program used to translate between ASCII and EBCDIC character sets for multiple operating system applications.

F-H

FEDEXNET

Federal **Ex**press **Net**work.

FTP

File **T**ransfer **P**rotocol. A TCP-based, application-layer, Internet Standard protocol for moving data files from one computer to another.

GISB

Gas **I**ndustry **S**tandards **B**oard. GISB serves as an industry forum for the development and promotion of standards that lead to a seamless marketplace for natural gas, as recognized by its customers, business community, participants, and regulatory entities. Employs HTTP protocol with pretty Good Privacy (PGP). See *PGP*.

I-L

IEBASE

(1) The batch front-end program used by EasyAccess2000 to communicate with IBM Interchange Services.

(2) IBM Expedite Base/AIX is a communications component of IBM Interchange Services for e-business that runs in the AIX Version 4.2.1 environment. Expedite Base/AIX is used to exchange electronic data with trading partners via Information Exchange, the mailbox component of IBM Interchange Services. IEBASE uses Comm-Press as its underlying compression and encryption software. See *IGN-I/E*.

IETF

Internet **E**ngineering **T**ask **F**orce. The body that defines standard Internet operating protocol such as TCP/IP and is supervised by the Internet Society Internet Architecture Board. Standards are expressed in the form of Requests for Comments (RFC). See *RFC*.

IGN-IE

IBM **G**lobal **N**etwork-**I**nformation **E**xchange. A mailbox component of the IBM Interchange Services used to used to exchange electronic data with trading partners. It uses the *IBM Expedite Base* (IEBASE) software as its communications component.

IP address

The four-byte address convention that uniquely identifies each node under Simple Network Management Protocol (SNMP). The format of the IP address is X.X.X.X, where X is one byte with a decimal value of 0 to 255. Users must define their own conventions for determining the IP address for the network they manage. See *SNMP*.

JCL

Job **C**ontrol **L**anguage. A language for describing jobs (units of work) to the MVS, OS/390, and VSE operating systems, which run on IBM's S/390 large server (mainframe computers). These operating systems allocate their time and space resources among the total number of jobs that have been started in the computer. Jobs in turn break down into job steps. All the statements required to run a particular program constitute a job step.

key pair

A private key and its corresponding public key. The public key can verify a digital signature created by using the corresponding private key. See *private key* and *public key*.

Lookup table file

A SecureManager2000 Security Runtime file containing records that define the security options being used between trading partners. Lookup table records contain keyword and values that define the (**1**) sender, (**2**) receiver, (**3**) data/transaction type, (**4**) security options, and (**5**) security structures for each trading partner relationship. Security options that may be specified in the lookup table include (**1**) compression, (**2**) encryption, (**3**) authentication, and (**4**) filtering.

M-P

MIME

Multipurpose **I**nternet **M**ail **E**xtension is an extension to the original Internet e-mail protocol that lets people exchange different kinds of data files on the Internet: audio, video, images, application programs, and other kinds, as well as the ASCII handled in the original protocol, the Simple Mail Transport Protocol (SMTP). Servers insert the MIME header at the beginning of any Web transmission. Clients use this header to select an appropriate "player" application for the type of data the header indicates. Some of these players are built into the Web client or browser (for example, all browser come with GIF and JPEG image players as well as the ability to handle HTML files); other players may need to be downloaded. New MIME data types are registered with the Internet Assigned Numbers Authority MIME is specified in detail in Internet RFC-1521 and RFC-1522. See *SMTP*.

MVS

Multiple **V**irtual **S**torage. CMS (Conversational Monitor System) is a product that comes with IBM's VM/ESA operating system and allows each of many simultaneous interactive users to appear to have an entire mainframe computer at their personal disposal. VM provides an extra layer of programming below an operating system, called the *control program* that handles the actual machine operation of the computer. The control program lets each operating system, such as MVS, appear to be in sole charge of the computer - effectively, creating a *virtual machine*.

participant

Reference to a trading partner in the SecureManager2000 application. See *trading partner*.

participant name

A program field that identifies the trading partner; normally the most commonly used name recognized for the trading partner, such as a surname, a system identification, etc.

passphrase

A string of 64 characters used to encrypt private keys. Passphrases (passwords) are randomly generated during the key generation process. They may be stored with the private key or written to a separate file when the SecureManager2000™ run-time files are imported.

PDS

Partitioned **D**ata **S**et. A highly structured IBM mainframe computer file that contains several named objects.

PFX

Prime **F**ile **T**ransfer.

POP3

Post **O**ffice **P**rotocol 3. A new standard that uses the Internet protocol to retrieve electronic mail from a server. This version can be used with or without Simple Mail Transfer Protocol (SMTP). POP3 mail servers are independent of the transport mechanism used to access them.

private key

The mathematical value of an asymmetric key pair that is **not** shared with trading partners. The private key works in conjunction with the public key to encrypt and decrypt data. For example, when the private key is used to encrypt data, only the public key can successfully decrypt that data. See *secret-key*.

Private Key file

A SecureManager2000 runtime file containing the private keys of local security participants that send secure data to outside trading partners. Private keys are never shared among trading partners. The private key file contains only those private keys that belong to local security participants that originate and send secure data from the site where Comm-Press2000 is executed.

Q-R-S

receiver

The receiving trading partner, system or process that is the destination of transmitted data.

RFC

Request **F**or **C**omment. Document series used as the primary means for communicating information about the Internet. Some RFCs are designated by the IEFC as Internet standards.

S/MIME - EDIINT

Secure/**M**ultipurpose **I**nternet **M**ail **E**xtensions. An Internet protocol [R2633, June 1999] to provide encryption and digital signatures for Internet mail messages.

secret key

The value used in a symmetric encryption algorithm to encrypt and decrypt data. Secret keys must be known only by the trading partners authorized to access the encrypted data.

SecureManager2000™

A bTrade.com product that manage key critical functions of a business-to-business electronic commerce network for customers. These include registering trading partners, classifying data, defining security relationships among partners, and distributing client software, SecureManager2000 is used to exchange and validate certificates or generate public/private keys for all trading partner participants. SecureManager2000 interoperates with public certificate authorities such as Entrust Technologies and Verisign, Inc.

sender

The sending trading partner, system or process that is the originator of transmitted data.

SMTP

Simple **M**ail **T**ransfer **P**rotocol. A TCP/IP protocol used in sending and receiving e-mail. However, since it's limited in its ability to queue messages at the receiving end, it's usually used with one of two other protocols, POP3 or Internet Message Access Protocol (IMAP), that let the user save messages in a server mailbox and download them periodically from the server. In other words, users typically use a program that uses SMTP for sending e-mail and either POP3 or IMAP for receiving messages that have been received for them at their local server. See *MIME*.

SSL

Secure **S**ockets **L**ayer. A program layer created by Netscape for managing the security of message transmissions in a network. Netscape's idea is that the programming for keeping your messages confidential ought to be contained in a program layer between an application (such as your Web browser or HTTP) and the Internet's TCP/IP layers. The SSL upper layer provides asymmetric cryptography for server authentication (verifying the server's identity to the client) and optional client authentication (verifying the client's identity to the server), and enables them to negotiate a symmetric encryption algorithm and secret session key (to use for data confidentiality) before the application protocol transmits or receives data. A keyed hash provides data integrity service for encapsulated data.

stored transfer

A predefined set of instructions used by EasyAccess2000 to control sending or receiving files between trading partners.

Symmetric Key file

A SecureManager2000 Security Runtime File containing the secret keys of local security participants that wish to send secure data to outside trading partners using secret key cryptography. The symmetric key file contains only those secret keys for data transfer relationships specified in SecureManager2000. The secret keys are stored in an encrypted format where unauthorized persons cannot view or use them outside the controlled environment. Comm-Press2000 decrypts the private key at execution time, then encrypts and authenticates the X12 EDI data.

T-W

TLS

Transport **L**ayer **S**ecurity. Btrade.com supports version 3 of this Netscape protocol. Secure Socket Layer Version 3.0 standard developed to provide security for web server and web browser applications. SSL has been endorsed and included in the Transport Layer Security protocol promoted with the Internet Engineering Task Force (IETF) by several major data communications technology corporations, such as IBM.

trading partner

A supplier, customer, service provider, or other party with whom business documents are routinely exchanged. Referred to as a *participant* in the SecureManager2000 application.

trading partner address book

A record of all trading partners' primary and mailbox information needed to send or receive a data transfer. If EasyAccess2000 cannot successfully complete the transfer after three attempts, EasyAccess2000 attempts to use the backup network and mailbox information.

transfer (stored)

A predefined set of instructions used by EasyAccess2000 to control sending or receiving files between trading partners.

UN/EDIFACT

United **N**ations rules for **E**lectronic **D**ata **I**nterchange for **A**dministration, **C**ommerce and **T**ransport. They comprise a set of standards, directories and guidelines for the electronic interchange of structured data related to trade in goods or services, between independent computerized information systems.

uncertified public key

The publicly disclosed component of a pair of cryptographic keys used for asymmetric encryption.

VAN

Value **A**dded **N**etwork. The source or service that resolves the issues resulting from communicating with a number of different trading partners. They provide EDI communication skills, expertise, and equipment necessary to communicate electronically.

X

X12

An international standard for EDI messages, developed by the Accredited Standards Committee (ASC) for the American National Standards Institute (ANSI).

X12.58

An ANSI security structures standard that defines data formats required for authentication and encryption to provide integrity, confidentiality, and verification of the security originator to the security recipient for the exchange of Electronic Data Interchange (EDI) data defined by Accredited Standards Committee (ASC) X12. See *ANSI ASC X12*.

Index